NOVIEMBRE 1986 ₳ 3,20 Nº20 REP. ARGENTINA



COMPUTACION PARA TODOS

100 Páginas

K-Test Con Premios

Aplicaciones Científicas en la Argentina

Software Para PC

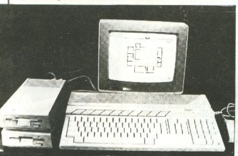
ATARI-MSX-CZ-TK

Archivos Comerciales Para las Commodore

Interfase Universal

Robot Controlado Por una C-64

SUMARIO



CARTA DEL DIRECTOR

El diseño y la construcción de un robot para la enseñanza y el entretenimiento, computadoras dedicadas a la biotecnología o la reconstrucción de procesos naturales, la interconexión entre universidades nacionales y extranjeras, son realidades de la informática nacional.

Estas aplicaciones científicas (tratadas en este número ¡que nuevamente tiene 100 páginas!) reafirman lo dicho por nosotros en cartas anteriores, acerca de la evolución de este sector.

Por otra parte, con equipos más grandes, la informática se instala definitivamente en establecimientos científicos y educativos. Su aplicación en el Museo de Ciencias Naturales de La Plata o en la Fundación Campomar, y la implementación en las casas de altos estudios indican que estamos en los umbrales del país "informatizado" acorde al tiempo que nos toca vivir.

CRISTIAN PUSSO





K64 COMPUTACION PARA TODOS

AÑO 2 Nº 20 NOVIEMBRE DE 1986

Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Secretario de Redacción
Ariel Testori

Prosecretarios
M.G. Verdomar Weiss (Coordinación)
Eduardo Mombello (Técnico)

Redacción
Pedro Sorop

Diagramación
Fernando Amengual
Tamara Migelson

Secretaria
Moni Ocampo

Departamento de Avisos
Oscar Devoto
Nelzo Capello

Departamento de Publicidad
Jefe: Dolores Urien
Promotores: Mónica Garibaldi
Edgardo López
Marita García

Servicios de Fotografía
Victor Grubicy
Eduardo Comesaña

K-64 es una Revista mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, Buenos Aires, Tel.: 46-2886 - 49-7130. Registro Nacional de la Propiedad Intelectual: 313.837 M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.

Imrpesión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica
Distribuidor en Capital: MARTINO, Juan de Garay 358, P.B. Capital, Tel.: 361-6962. Distribuidor interior: DGP, Hipólito Yrigoyen 1450, Capital, Tel.: 38-9266/9800.
K-64 ISSN 0326-8285. Los ejemplares atrasados se venderán al precio del último número en circulación.

CORREO ARGENTINO CASA CENTRAL | FRANQUEO PAGO CONCESION Nº 2538 | TARIFA REDUCIDA CONCESION Nº 636

Miembro de la Asociación Argentina de Editores de Revistas



¿POR QUE UN MONITOR?

TALENT DM-120M es el nombre del nuevo monitor de fósforo verde destinado a completar el equipo de hardware que se enlaza a los tentáculos de la computadora Talent MSX.



Encerrado en un magnífico gabinete color tiza, de diseño moderno y líneas futuristas, el monitor es de 12 pulgadas. En su lateral derecho posee un parlante encargado de reproducir el sonido que sale del conector de SOUND de la computadora y se une con el de AUDIO IN del aparato.
ON-VOL, CONTRAST y BRIGHTNESS son tres potenciómetros que ubicados en el frente, están destinados al encendido y volúmen, contraste y brillo del nuevo "chiche" de Telemática.
Los acompaña un delicado led rojo, indicador del encendido.
Se conecta directamente a la líneas de 220 volts y su consumo es de 130 miliamperes, lo que produce una potencia de 30 watts.
Si bien un televisor color ofrece el atractivo de una gama de 16 colores de nuestra computadora, la calidad del mismo, en cuanto al tubo de rayos catódicos, es baja y no adecuada para alta resolución de computadoras.

CURSO

La Universidad Nacional del Centro realizará la segunda escuela Brasileño-Argentina de Informática —EBAI—. El curso se dictará entre el nueve y el 22 de febrero del año próximo. Para mayor información sobre becas e inscripciones hay que dirigirse a la oficina de la EBAI, Suipacha 760, quinto piso, de nueve a doce.

LA GASTRONO MIA COMPU TARIZADA

Hace cinco años, muy pocos hubiéramos imaginado encontrar lugares de comida con computadores. Ahora, la empresa Mesicom S.A. presenta un plan capaz de convertir la ficción en realidad.
El S.A.C. permite almacenar listas de precios de hasta 1000 productos en varios idiomas y con la posibilidad de cambiar



los valores de los artículos en forma manual, automática o por grupo.
El equipo para este sistema abarca una consola MSX, su disquetera, una impresora y un televisor o monitor.
Con S.A.C. se lleva el control de hasta 100 mesas con 60 cargos cada una comprendiendo la adición y el control de mesas abiertas, mozos y servicios.

LA AVANZADA MSX DE TOSHIBA

La Toshiba HX-20 es una computadora de características avanzadas. De segunda generación, su capacidad de memoria total es de 144 KBytes con una muy novedosa función. Presenta el "Ram-Disk" y Basic Ampliado, que le permite almacenar datos en Ram, en 32 KBytes adicionales. Por otra parte dispone de un programa completo para procesamiento de texto incorporado a la misma máquina accediendo a través del menú principal. Como periféricos, Toshiba ofrece la impresora de puntos de alta velocidad, plotter de cuatro colores, unidad de microdiscos de formato vertical, datasette, joysticks, interface RS232C para comunicación telefónica y un teclado sintetizador.

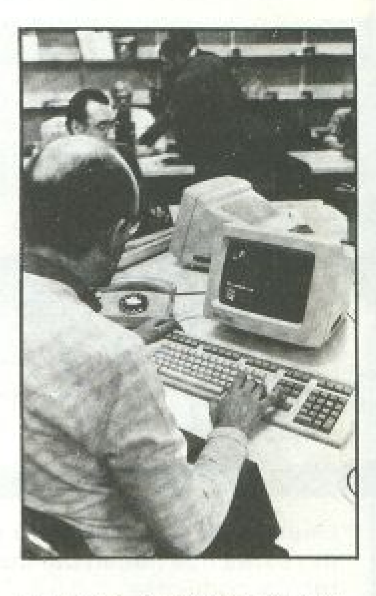
ADJUDICACIONES

Las empresas FIMPAR y el Centro Instrumental de Rosario fueron beneficiados por las primeras adjudicaciones en el sector informático. FIMPAR se presentó para el rubro periféricos mientras que la otra empresa lo hizo para productos complementarios. También se otorgarían créditos fiscales a los proyectos de Sisteco, CNL-BULL y Microsistemas, pero en rubros de mayor relevancia.

CENTRALIZADOS VS. DISTRIBUIDOS

"Destacados expertos extranjeros han confirmado que los sistemas distribuidos se imponen por razones tecnológicas y económicas sobre los centralizados" expresó el subsecretario de Informática y Desarrollo, doctor, Carlos María Correa, al referirse al panel realizado sobre el tema en las 16ª Jornadas de Informática e Investigación Operativa. En el panel participaron, entre otros, los expertos internacionales Douglas Jensen, de la Universidad Carnegie-Mellon de EE.UU.; Mandred Broy de la Universidad de Passau de Alemania Federal; y Norma Lijtmaer, de la Universidad de Piza.
"Las posiciones de los especialistas —concluyó Correa— coinciden plenamente con el proyecto de política informática esbozado por la Comisión Nacional de Informática y puesto en marcha por la subsecretaría de Informática y Desarrollo".

BASE DE DATOS: DEMOSTRACION GRATUITA



DELPHI, el sistema de información y comunicación en línea que incluye base de datos, ofrece a todos aquellos que posean un modem CCITT y por supuesto, una computadora, la posibilidad de conectarse con ellos en forma **gratuita** para dar un paseo por sus opciones informáticas. Además, una vez conectados, y luego de haber sido sorprendidos por la inmensidad de alternativas que DELPHI nos ofrece, tendremos la posibilidad de ins-

cribirnos directamente desde nuestro hogar y en la misma comunicación. Para lograr todo esto bastará enlazar nuestra fiel computadora al número telefónico 33-5188 de DELPHI y responder al mensaje "USER NAME:" con el que le agrada a este mainframe: DELPHI.

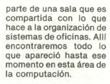
PRESENTACION DE CeBIT

Fueron presentadas alrededor de 20 ferias que se realizarán en la ciudad de Hannover, Alemania Federal, el próximo año. Algunas de ellas tendrán como protagonistas los últimos desarrollos en Informática y Robótica. Entre las muestras apuntadas encontramos la de CeBIT (Centro Mundial de las técnicas de Oficina, Informática y Comunicación) a desarrollarse entre el 4 a 11 de marzo de 1987.



En la conferencia de prensa ofrecida, el señor Sepp Heckmann, miembro del directorio de la Sociedad Ferial de Hannover, recordó el éxito obtenido en las reuniones internacionales anteriores, "CeBIT —dijo— incrementó entre 1985 y 1986 el número de sus visitantes en un 60 por ciento, convirtiéndose en este primer año de su realización independiente en la feria de comunicación más importante a nivel mundial".

En cuanto al tema de microcomputación —tema que nos toca de cerca— se le destinarán tres salas completas de las dieciocho con las que cuenta la feria, sumado a eso parte de una sala que es compartida con lo que hace a la organización de sistemas de oficinas. Allí encontraremos todo lo que apareció hasta ese momento en esta área de la computación.



ODONTOLOGIA Y COMPUTACION

Una empresa de informática desarrolló soft sobre la norma MSX para incorporar a los odontólogos a la gigantesca ola de avances técnicos.

Se trata de soft para controlar un registro detallado de citas de pacientes, con indicación en pantalla de totales diarios y semanales. Pero más allá de la actividad administrativa y contable, que por supuesto es más amplia, la gran novedad se da en el campo específico. Mediante este sistema se puede obtener una representación por pantalla de las piezas dentales de cada paciente, registrar el estado de cada pieza y el resultado del tratamiento efectuado.

El Sistema ODONTOPACK fue desarrollado por HIPOCAMPO INFORMATICA S.R.L., bajo supervisión del Dr. Eduardo Zabotinsky.

NUEVO MODEM COMMODORE COMPATIBLE



Infotel S.A. (representante en Argentina de THE SOURCE y COMPUSERVE) ha lanzado al mercado un MODEM COMMODORE COMPATIBLE, el INFO-300, que le otorga a los usuarios de commodore un amplio horizonte en las comunicaciones con servicios de información nacionales e internacionales como THE SOURCE y COMPUSERVE, boletines de acceso gratuito, etc.

Este modem se instala directamente en el port del usuario, es Binorma (bell y ccitt), posee discado manual o automático; respuesta manual o automática. Su carcaza es metálica lo que le otorga una gran solidez.

Las velocidades en que opera van de 0 a 300 Bps Full Duplex.

El desarrollo de este equipo es íntegramente realizado en el país.

DISQUETERA PARA LAS CZ-SPECTRUM

Se trata de OPUS DISCOVERY, una disquetera que ya tiene su trayectoria en el mercado mundial.

La compañía CZERWENY ha comenzado a fabricarla en la argentina y ya es inminente su aparición en los comercios.

Todos aquellos que se preguntaban para qué servían las instrucciones que se encuentran en la parte superior del teclado (CAT, FORMAT, etcétera) encontrarán la respueta en el manual de esta gran disquetera.

Esta unidad posee además un conector para Joystick tipo Kempstone dándonos la posibilidad de acceder a los juegos más famosos de esta máquina, con el afamado palito juguetón.

Posee además un pórtico paralelo que nos permitirá conectar cualquier impresora con esta entrada característica.

La misma utiliza discos de 3 pulgadas y media envasados en estuches rígidos de material plástico. Cada disco podrá soportar una capacidad de 180 Kbytes.



MINI-CHIPS

El descubrimiento de un microscopio que utiliza la "técnica del barrido por el efecto de túnel" fue motivo para la entrega del premio Nobel de Física y de Química 1986. Los galardonados, Gerd Binning y Heinrich Rohrrer, pusieron

a punto el aparato en los laboratorios de la empresa IBM en Zurich. El microscopio permite examinar la estructura superficial de un material —átomo por átomo—. Según la Academia el método desarrollado podría ayudar a la creación de "chips" más pequeños para computadoras.

CONCURSO

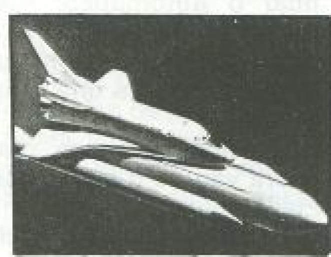
Se abrió un concurso de ensayos sobre el tema "Perspectivas de la transferencia electrónica de fondos —EFT—". El mismo está dirigido a estudiantes universitarios de América Latina. El ganador recibirá un premio de tres mil dólares y un viaje a Caracas, Venezuela. Los auspiciantes son NCR y la Comisión Permanente EFT.

PROYECTOS

La subsecretaría de informática y desarrollo y la oficina intergubernamental para la informática —IBI—, se refirieron a proyectos del área. Los adelantos se darán en el ámbito judicial, industrial, educativo y de recursos humanos. En 1987 el IBI prevé continuar con aportes para la política nacional de informática, la microelectrónica, universidades.

LA "GUERRA DE LAS GALAXIAS"

"Los programas de computación que se necesitan para manejar el sistema de defensa espacial "Guerra de las Galaxias", aparecen más allá del alcance de las capacidades de programación conocidas o contempladas". Así lo afirmó en su artículo John Tirman, de "Los Angeles Times". Parecería que el mal paso dado por el presidente Reagan en Reikiavik en relación a este proyecto llega en un momento de escepticismo sobre la factibilidad y conveniencia de dicho programa.
Las barreras técnicas, los costos y la vulnerabilidad de la "Guerra de las Galaxias" ahora son mejor comprendidos por los científicos entre los que empieza a manifestarse cierto descontento. Siete mil hombres de ciencia norteamericanos ya expresaron su "fuerte oposición" al proyecto del "escudo espacial".
Un profesor de Física de la Universidad de Illinois, John Kogut, es uno de los líderes. Kogut considera que el plan de defensa espacial es un obstáculo



para el control de armamentos y lo calificó de "desestabilizante, peligroso y técnicamente irrealizable".

DOS NUEVAS QL

Dos nuevas versiones de la Sinclair QL serán lanzadas en Inglaterra en los próximos meses.
La primera será fabricada por CST, una empresa que fabrica drives de disco. Usará un 68000, y estará equipada con una unidad de disco de 720 k. Será compatible con todos los periféricos de la QL original y aun no se sabe si será 100 por ciento compatible en software. Estará equipada con un teclado tipo IBM y su precio será de alrededor de 550 libras.
Una versión plus de esta máquina (que se comercializará bajo el nombre de THOR) incluirá un disco duro de 20 MBytes.
Por último, la empresa CTS planea introducir al mercado un modelo equipado con un microprocesador 68020 de Motorola, con una longuitud de palabra de 32 bits. Esta, permitirá trabajar en modo multiusuario, aunque aun no hay idea de costos.
Por otra parte, otra neva QL será comercializada bajo el nombre de QLT. Será fabricada por Farmintel, el brazo inglés de la empresa Italiana de hardware Sandy.
También tendrá un 68000 como procesador central, será compatible un soft con la QL original y su velocidad de procesamiento será de 3 a 10 veces mayor. Estará equipada con un mínimo de 0,5 Mbytes de RAM expandible a 8 Mega, capacidad para conectarle dos unidades de discos blando y una de disco duro. El precio será inferior a las 500 libras.

MICROS

IBM lanzó una estrategia para contrarrestar la ofensiva de Digital Equipment Corporation. La conocida empresa prepara una familia de cuatro microcomputadoras para el mundo de los negocios. Por otra parte, la compañía IBM anunció una reconsideración en el precio del Software en el campo mayorista.

MINITEL

Trasmitir chistes de judíos, por lo general malos, el texto completo de la Declaración de los Derechos Humanos de la Revolución Francesa, promover la medicina natural combinada con dietas naturistas y baños de algas o reproducir las partidas de ajedrez entre Karpov y Kasparov. Estos son algunos de los 3000 servicios que presta Minitel de Francia. Posiblemente ese sea el secreto del éxito del videotexto entre los galos mientras en el resto del mundo el servicio empieza a perder interés. Es que casi dos millones y medio de hogares y oficinas francesas cuentan con las terminales de Minitel. Su uso se hizo tan cotidiano que hasta las canciones de moda para jóvenes ya hablan de un adolescente que se enamora de una chica a través de su computadora. ¿Ficción que se hace realidad? ¿Alienación versus tecnología? Todo se hace posible cuando tenemos el futuro a la vuelta de una esquina.

ELECTRONIA '86

Durante las II Exposición y Simposio de la Industria Electrónica Argentina, realizadas en el Buenos Aires Sheraton Hotel, el Programa Nacional de Informática y Electónica —PNIE— de la Subsecretaría de Informática y Desarrollo presentó equipos pertenecientes a los proyectos que éste subsidia.
El PNIE presentó entre otras actividades una memoria de dígitos osciloscopio, realizado por la Universidad Nacional de Rosario; un transceptor en la banda de microonda, de la Universidad Nacional de Tucumán; y una interfase para controlar máquinas de escribir electrónicas, también de la Universidad Nacional de Tucumán.
Asimismo, se ilustró la actividad desarrollada en la Dirección de Planificación y Coordinación del Gobierno de la Provincia de Entre Ríos, mediante la descripción del proyecto "Sistematización de la Información de Bases".

UN PLAN DE TALENT

Talent creó un nuevo sistema de ahorro previo para que nadie quede sin la posibilidad de contar con una computadora MSX.
Este nuevo plan en realidad son tres y cada uno con diferentes opciones.
El Plan 1 ofrece una computadora MSX DPC-200 (alternativa Unidad de Disco Flexible Talent MSX DPF-550 ó 555 sin diferencia de precios). El Plan 2 estipula la computadora y la disquetera, y como alternativa el monitor Talent 14" y la diferencia para cancelar cuotas.
El Plan 3 abarca la computadora, la disquetera y el monitor.





LIBROS

MSX. GUIA DEL USUARIO

AUTOR: PAUL HOFFMAN
EDITA: OSBORNE/MC GRAW-HILL
DISTRIBUYE: CUSPIDE



Si poseemos una MSX, o la pedimos prestada y buscamos desesperadamente información detallada de cada una de las instrucciones y comandos de esta norma, si queremos saber todo lo que se puede hacer con esta maquinúscula y si necesitamos mapas de memoria, variables del sistema operativo, direcciones de memoria útiles, manejo de rutinas en código de máquina o una introducción para hacer el primer programa Basic, entonces, sólo entonces compremos este libro.
En más de 290 páginas de excelente edición, se propone hacernos saber todo lo que se refiere a las MSX, y desgraciadamente lo logra.

DESCUBRE TU MSX

AUTOR: JOE PRITCHARD
EDITA: ANAYA

"Descubre tu MSX" nos ayudará a adentrarnos en el conocimiento del control de los monstruos más horrendos de los que habitan detrás de los teclados de las computadoras de esta norma: el VDP, el PSG y el PPI.
Por supuesto, el autor, en forma clara e ilustrativa, nos explicará cómo es que interactúan éstos con la CPU, que como todos sabemos es el Z-80.
Además veremos aquí algunas aplicaciones útiles de los comandos más importantes y su comportamiento.
El texto ofrece una clara ejemplificación del manejo de variables a nivel sistema operativo y Assembler.
Podemos pues tenerlo como un libro de consulta

PLANILLA ELECTRÓNICA INTEGRADA

AUTORES: PATRICIA SEYBOLD-LINDA O'KEEFFE
EDITA: MC GRAW HILL
DISTRIBUYE: CÚSPIDE



En planillas electrónicas para PC nunca está dicha la última palabra.
Este libro se encargará de esclarecer el significado, sus posibles aplicaciones y dar gran variedad de ejemplos de cada una de las órdenes que el LOTUS 1-2-3, Context MBA y Symphony poseen.
Quizás algunos lectores se pregunten que es el Symphony. Se trata de un Super LOTUS 1/2/3 que posee manejo de ventanas con efecto "zoom", una ampliación considerable de los comandos del anterior y aumento de la velocidad de trabajo entre algunas de sus novedades.
Aunque este desarrollo no estaba terminado cuando fue escrito este libro, se comentan experiencias previas de los encuentros de las autoras con este magnífico y complejo utilitario.

GUÍA DE COMPATIBLES DEL IBM/PC

AUTOR: MELODY NEWROCK
DISTRIBUYE: CÚSPIDE
EDITA: MC GRAW HILL



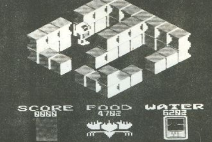
Esta es una guía detallada de las PC que más trayectoria han tenido en la historia de las compatibles.
Seguramente, a muchos lectores les queda la duda sobre qué es la compatibilidad y cómo está implementada ésta en los modelos más afamados.
Este libro nos aclara todas las dudas al respecto, haciéndose muy útil a la hora de la elección entre las distintas máquinas.
Posee explicaciones claras sobre modelos que van desde la PERSONAL AT&T Modelo 6300 hasta la Profetional de WANG. Pasando por las Corona, Profesional TI, Panasonic Senior Partner y Televideo TS 1605, por nombrar algunas.
Incluye, además fotos y diagramas que esclarecen los comentarios que ocupan casi 222 páginas.

MACINTOSH APLICACIONES DE MULTIPLAN Y MAC PAINT

AUTOR: EDDIE ADAMIS
DISTRIBUYE: CÚSPIDE
(REPRESENTANTE DE MC GRAW HILL)



El presente libro será de gran utilidad para aquellos que posean la legendaria Apple en su versión Macintosh.
Quizás muchos usuarios de Commodore conocen ya los utilitarios: Geos, print Shop Companion, etcétera, pero lo que quizás no saben es que éstos tienen un padre en común que ya hace bastante tiempo habita en las grandes computadoras.
Este es el caso del Mac Paint, un revolucionario utilitario que lideró bastante tiempo el desarrollo en base a Incons o figuras y Windows o ventanas.
El mismo sistema de explicación es el usado por el Multiplán, que como ya algunos sabrán, es una planilla electrónica que ha dado que hablar, y que ha pasado ya por las IBM PC y todas sus compatibles, convirtiéndose en un clásico.



Así es que esta publicación de 140 páginas, con excelentes gráficos y pantallas nos introduce en el uso y aplicaciones de estas increíbles herramientas.

SOFTWARE

KNIGHT GAMES

COMP: C-64
CLASE: ENTRETENIMIENTO
SOPORTE: DISCO
DISTRIBUYE: DATA-GAMES

Si nuestra ilusión frustrada es haber pertenecido a la dulce y bélica Edad Media, este juego es el nuestro.




Podremos obtener ocho personificaciones distintas, para amasijarnos contra nuestro amigo o computadora.
Como siempre en estas máquinas la calidad gráfica y sonora son casi insuperables.
Podremos así, batirnos a duelo como hacheros, con las pinchudas esferas encadenas, o competir en ballesta o como arqueros, entre otras cosas. Todo esto acompañados de música al mejor estilo Hollywood y paisajes alucinantes.

CHIMERA

COMP: C-64
CLASE: ENTRETENIMIENTO
DISTRIBUYE: DATA-GAMES
SOPORTES: CAS/DISK

Nuestro robótico subconciente se verá reflejado en un gigantesco laberinto compuesto de infinidad de habitaciones.
Podríamos decir que es una diversión en la que la parte artística tira más hacia el cubismo que el romanticismo. Además, dentro de esta rama pictórica los efectos visuales son excelentes al igual que el sonido.
Por si a alguno le queda dudas de cómo se pronuncia la palabra CHIMERA en inglés, el programa al comenzar, con suave y romántica voz, nos lo pronunciará. Casi podríamos decir que la voz es tan suave como el sonido que emite el robot cuando es destruido.

PRINT SHOP Y PRINT SHOP COMPANION

COMP: C-64
CLASE: UTILITARIO
SOPORTE: DISK

Dos utilitarios de lujo. Están orientados al diseño gráfico en general y a su posterior representación visual en la impresora.
La última versión (COMPANION), es totalmente compatible con la primera, el PRINT SHOP.

Con ellos podremos diseñar calendarios, tarjetas de invitación ilustradas con cualqueira de las figuras del banco de imágenes que los acompaña, carátulas enmarcadas o crear cualquier figura por medio de un excelente editor de gráficos.



En suma, es una gran herramienta al servicio del artista gráfico.

GEOS

COMP: C-64
CLASE: UTILITARIO
SOPORTE: DISK

Graphic Environment Operating System, es un paquete de programas utilitarios en los cuales hay desde un editor de textos con varios tipos de letras hasta un editor de gráficos de alta resolución.

Basado en el sistema de Icons o figuras representativas y ventanas o Windows, su manejo es extremadamente sencillo y se asemeja mucho al Mac Paint de Apple.

Hallaremos una calculadora de acceso permanente al igual que el reloj y su alarma programable.

En síntesis, este pack, será tan útil como lo es el "News Room" pero orientado más hacia el trabajo gráfico de escritorio.

DUAD

COMP: MSX
CLASE: UTILITARIO
SOPORTE: DISK

DUAD es un editor, compilador, ensamblador/desensamblador y seguidor de programas Assembler del Z-80.

Es realmente completísimo, y hace que trabajar en este lenguaje sea un placer.

Permite obtener ayuda en todo momento, por medio del comando help, que

```
LISTADOS DE LOS ARCHIVOS *.HLP (HELP) DEL DUAD


    Assembler command summary

    *MENU     - exit to MENU
    *EDIT     - go to EDITOR
    *DIS      - go to DIS-ASSEMBLER
    *DEBUG    - go to DEBUGGER (Full)
    *SDEBUG   - go to DEBUGGER (Small)
    *BASIC    - return to MSX-BASIC
    *RST      - re-do from start
    *HELP(?)  - show this message
    *DIR      - list directory

Command summary

A       - one line assembler
BS      - break point set
BR      - break point reset
BL      - break point list
DP      - memory dump by page
D       - memory dump
E       - edit memory
S       - search string from memory
F       - memory fill
M       - memory move
MAP     - list of memory map
LP      - dis-assembler by page
L       - dis-assembler
G       - go
T       - trace
H       - trace and not information
I       - ignore of trace
U       - ignore and not information

Hit any key!!
```

puede abreviarse con un signo de interrogación.

Si no conseguimos el manual del mismo, podremos obtener un buen apoyo, listando por impresora todos los archivos HLP que acompañan al mismo.

La calidad de éste está superasegurada, ya que los creadores del mismo pertenecen a la firma ASCII, que seguramente les suena de algún lado ¿no? Cada módulo del programa está grabado por separado, y podremos tener en el menú principal las siguientes, opciones: ASSEMBLER, DIS-ASSEMBLER, SCREEN EDITOR, DEBUGGER FULL, DEBUGGER SMALL, BASIC y HELP.

Todas sus opciones son super potentes. Como muestra podemos decir que un programa fuente puede generar: un archivo ejecutable, uno listable con los códigos hexadecimales y sus posiciones de memoria, un archivo de niveles o etiquetas por orden alfabético, un archivo de etiquetas por orden de aparición o un Cross Referens o listado de las líneas en las cuales son definidas o llamadas las etiquetas o niveles.

En suma, de lo mejor.

ARTIST

COMP: MSX
CLASE: UTILITARIO
SOPORTE: DISK

Todos aquellos que gustan del arte de dibujar o diseñar pantallas, podrán hacerlo de la mano del ARTIST. Este programa con millones de opciones, creado por YAMAHA MUSIC CORPORATION, nos permitirá crear a nuestro antojo.

Podremos incluir texto con caracteres tipo Sheriff o modernos de tamaño 16 por 16 pixels. Posibilita dibujar con varios trazos o aerosoles, hacer todo tipo de figuras completas o no, pintar, tramar eligiendo una trama del banco realizado para las mismas, dibujar pentagramas con todas sus figuras y casi todo lo que se nos ocurra.

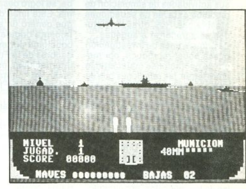
Como vemos se está elevando la calidad del software para todas las máquinas en general, adaptándose a los nuevos medios de hard. Por esto es posible trabajar tanto con un mouse como con las teclas de cursor.

También podremos grabar como archivos las pantallas que diseñemos, tanto en disco como en cassette, o sacarlas por una impresora MSX (no cualquiera).

Vale la pena.

BEACH HEAD

COMP.: CZ SPECTRUM / SINCLAIR
SOPORTE: CASSETTE
CLASE: JUEGO



Este entretenido pasatiempo se trata de una batalla para conseguir desembarcar en territorio enemigo.

Estamos a bordo de un barco equipado con cañones que nos permitirán atacar a los aviones que intentan destruirnos.

Al frente hay barcos y algunos son portaviones que al principio aparentarán ser inofensivos. Pero de repente comenzarán a moverse y a lanzarnos bombas.

Los cañones pueden subir o bajar para facilitar el tiro a los aviones y a los barcos.

Hay helicópteros que pasan de vez en cuando y si conseguimos derribarlos, seremos premiados con varios puntos.

Pero la operación de ataque no es solamente en el mar. Al pasar de nivel las cosas se complican porque tendremos que desembarcar y atacaremos desde un tanque. Este transporte pasará por campos minados que deberemos esquivar. El territorio del enemigo estará lleno de trincheras y de bases.

Al comienzo del programa nos ofrecen las instrucciones que habría que leer si nos sentamos por primera vez a entretenernos con este juego.

Cuando el programa se haya cargado, si no presionamos ninguna tecla, aparecerá una demostración que nos servirá para entender cómo se juega y saber con qué enemigos nos enfrentaremos.

Los gráficos de los barcos, aviones y tanques son muy buenos y la sensación de movimientos está lograda bastante bien.

No será fácil poder integrar la tabla de score, pero vale la pena intentar pertenecer a esta lista de personas que seguramente desperdiciaron mucho tiempo frente a este juego (igual que lo vamos a hacer nosotros).

VALIOSOS DESARROLLOS ARGENTINOS DE SOFTWARE

Entre tubos de ensayos y mecheros, entre estufas y caldos de cultivo, las computadoras de la Fundación Campomar se dedican a la biotecnología.

Este instituto, que dirige el premio Nobel Luis Leloir, es un centro de investigación científica en biología. Allí, en el laboratorio 104, Daniel Gluschankof, licenciado en matemáticas, y el doctor José Manuel Olavarría, químico, desarrollan programas de computación que sirven a esa tarea.

En sus inicios (unos cuatro años atrás), el laboratorio comenzó sus actividades de simulación de modelos a través de ecuaciones algebraicas. Hoy, otras inquietudes se agregaron a aquéllas.

La primera línea de investigación busca poder reproducir, por medio de operaciones matemáticas entre letras, las relaciones que se entablan entre diversos elementos químicos.

El laboratorio trabaja con enzimas, o sea aquellos productos que provocan la catálisis de un proceso químico, que hacen posible su realización. Una enzima (o, en realidad, varias) es la que permite que un asado se transforme en alimento para las células de los músculos, por ejemplo. Estos "catalizadores" reaccionan de determinada manera en presencia de algunos productos: las ecuaciones que nuestros investigadores buscan deben representar esas posibles reacciones.

Para eso, primero trazan un diagrama de transiciones, donde figuran, a la manera de un mapa, todas las posibles relaciones entre los elementos presentes. De allí se deduce un modelo, y la computadora es la encargada de traducirlo a una ecuación algebraica que lo represente. En verdad, suena sencillo; el problema era desarrollar el software para que la computadora lo hiciese.

Lo interesante de poder representar ese modelo por medio de una ecuación algebraica (o sea que se trata de letras, no de números) es que se puede aplicar a cualquier caso concreto, con sólo reemplazar las letras por los valores numéricos.

Pese a lo dificultado que implicaba, se logró desarrollar el programa. Esta línea de investigación, que generó un software absolutamente original, fue seguida y concluida por estos dos investigadores de la Fundación. A nivel internacional, esto se debe destacar, es "punta", o sea que es vanguardia en la tecnología de investigación.

El doctor Olavarría destaca otro aspecto. "Lo interesante —dice— es que nosotros archivamos las ecuaciones que representan los diversos modelos, las que pueden luego ser utilizadas en otros casos". Tienen idea de formar, así, un singular banco de datos donde archiven ecuaciones algebraicas que simulen modelos enzimáticos.

Sin el auxilio de las computadoras, este trabajo (haciéndolo "a mano") sería tarea de años y años. Pero, aparte de la velocidad con que la computadora resuelve la ecuación, el software desarrollado también provee la capacidad de comparar dos modelos (o ecuaciones) entre sí, para poder encontrar similitudes y diferencias entre ellos.

El conocimiento científico, nos han insistido tantas veces nuestros profesores, requiere demostración. Y los modelos enzimáticos no escapan a la regla. Una vez que la computadora elabora el modelo, los investigadores llevan el caso a la realidad concreta. Simplificando la cosa, toman un tubo de ensayo y ponen dentro de él todos los componentes del modelo en la cantidad señalada y controlan si la reacción, los productos de ésta, la velocidad con que se produjo, etc. son los esperados. De no serlo, el modelo no representa exactamente la realidad y debe ser ajustado.

Respecto del lenguaje que se usa para trabajar en este tipo de aplicación de la informática, Daniel Gluschankof nos dice que "en un principio trabajamos con Basic, pero luego nos fue insuficiente, así que seguimos con el Pascal, al que le hicimos algunas mejoras".

Luego de casi cuatro años de trabajo, el programa está concluído. "Funciona —dicen los investigadores—, pero todavía le hacemos algunos ajustes".

Una particularidad de este sistema de estudio de modelos es que está preparado para que lo opere gente que no esté capacitada sobre el manejo de las computadoras, sino que ha sido diseñado de tal manera que pueda ser utilizado por investigadores que se especialicen en enzimología (la que, obviamente, es la disciplina que estudia las enzimas).

La experiencia realizada en el laboratorio 104 es, sobre todo, la búsqueda de una metodología de estudio de modelos. Por eso, dice el doctor Olavarría "podría trasladarse al estudio de otros tipos de modelos, como puede ser

el estudio del tráfico urbano, o la recolección de basura".
Sin embargo, los principales logros de esa línea de investigación, para este científico, es que se consiguió "desarrollar ecuaciones que representen cualquier tipo de diagramas, que esas ecuaciones pasan a formar parte de un archivo y que luego se puede acceder a ellas para volverlas a usar".
El laboratorio cuenta con dos sistemas de computación. Un sistema de desarrollo en base Bus S 100 de 48 K: "un equipo de baja potencia, característico de los años 79-80", dice Daniel. El otro equipo es una mini-computadora IBM Series 1, prestada por esa empresa, que tiene 192 K, un disco fijo, disketera, dos terminales y dos impresoras IBM 4974. "Si bien es una máquina que tiene casi diez años —dice el matemático—, y que tiene menos potencia que algunas PC más modernas, se adapta muy bien al trabajo que nosotros estamos realizando".

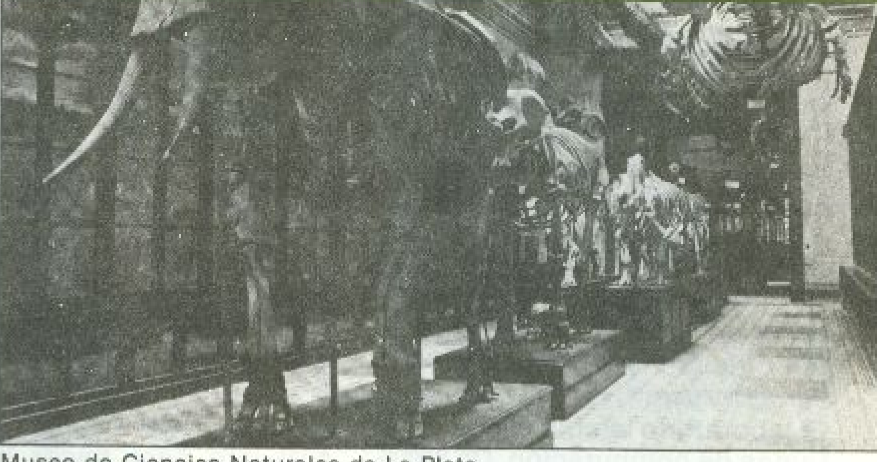
Células congeladas y nuestros parientes los monos

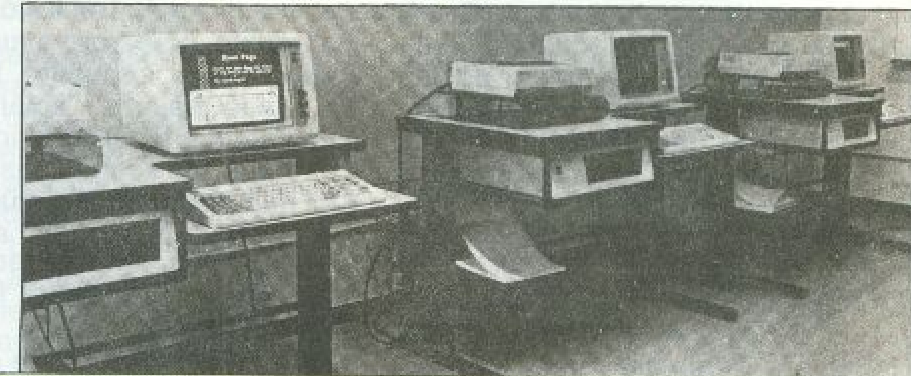
Cuando ese primer software comenzó a ser operativo, el laboratorio 104 empezó a buscar en otras líneas de investigación. Uno de los programas desarrollados tiene que ver con el estudio de material genético en proteínas.
"El programa para ese tipo de estudios existía ya en otros países, pero no lo teníamos ni se adaptaba a nuestras máquinas", dice Daniel. Así que, partiendo de lo que sabían sobre el programa


El doctor Olavarría operando la computadora

existente, desarrollaron todo el software para hacer lo mismo que hacía aquél. Les llevó un año de trabajo, pero hoy ya funciona, "aunque, por supuesto, todavía se puede mejorar", aclara.
Esencialmente, el soft sirve para comparar las secuencias de ácidos nucleicos (ADN). Estudiándolas se puede determinar a qué tipo de proteínas pertenece, analizar cómo reaccionará frente a una enzima, etc.
"Este programa permite comparar dos secuencias de ácidos nucleicos para saber qué posibles relaciones hay entre ellas", dice Daniel. "Si comparásemos secuencias de ADN de una proteína humana con la de un chimpancé - -ejemplifica—, encontraríamos que existe un 96 por ciento, aproximadamente, de similitud, con lo que se demuestra que dentro de la escala de la evolución somos casi parientes".
Este investigador nos cuenta que en EEUU existe un banco de datos con 5.000.000 de caracteres de extensión donde se acumulan las secuencias de ADN que ya se han estudiado. "Cuando podamos, vamos a establecer una conexión de nuestra computadora con ese banco, para facilitar la tarea de comparación", agrega. Y hay otra línea de trabajo para las computadoras del laboratorio 104: el archivo de los datos de las células que se encuentran congeladas en los depósitos de los laboratorios.
Cuando los investigadores consiguen algún tejido celular que es particularmente interesante, o que desean conservar por alguna razón, se lo guarda en termos con nitrógeno líquido (a una temperatura de menos de −200 grados centígrados). Existen cientos de esos tejidos congelados: archivar los antecedentes de cada uno (de dónde viene, desde cuándo está allí, si se lo sacó para reproducirlo, las condiciones en que se lo ha conservado, etc.) sería imposible sin el auxilio de las computadoras. Allí el laboratorio 104 ha concurrido en auxilio de los otros investigadores de la Fundación y ha desarrollado un programa que funciona como banco de datos de esos antecedentes.
Esto es lo que se hace dentro del laboratorio 104: científicos y computadoras dedicados al mundo de la biotecnología.

"EN SEGUNDOS, VEMOS PASAR MILLONES DE AÑOS"


Museo de Ciencias Naturales de La Plata



Con computadoras personales, científicos argentinos pueden reconstruir procesos de la naturaleza, organizar colecciones, realizar bancos de datos y, algún día, quizás hallen el "eslabón perdido..."

¿Qué grado de parentesco existe entre un ombú y una margarita? ¿Podrán descender de un mismo ancestro común?

Los caballos que actualmente habitan el campo argentino: ¿son descendientes de aquellos equus que poblaban las llanuras americanas hace 3 millones de años? Responder estas preguntas forma parte, también, del quehacer de una computadora. O, por lo menos, de las que trabajan en el Museo de Ciencias Naturales de La Plata.

Estas máquinas, aparte de ser exactos instrumentos de cálculo, de realizar el procesamiento de la palabra, de hacer la gestión de un banco de datos, simulan procesos naturales.

En la naturaleza existen procesos que se llevan a cabo a través de muchísimos años: el desarrollo de cierta capacidad, la especialización en una función, la aparición de algún caracter específico en un animal. Son procesos que se llevan a cabo durante enormes períodos de tiempo. Con el uso de las computadoras, es posible simular esos procesos para su estudio y, en segundos, ver pasar millones de años.

"Con las computadoras podemos reconstruir procesos históricos, como ser la historia de una familia de plantas", dice el doctor Jorge Crisci, director del proyecto recientemente suscripto por la Facultad de Ciencias Naturales y el Museo de la Universidad Nacional de La Plata con IBM Argentina.

Crisci y sus colaboradores, con una serie de programas que simulan las posibles relaciones entre dos especies vegetales, analizan la posibilidad de que entre ellas existe algún "parentesco".

"La naturaleza por lo general recorre los caminos más cortos en la evolución", dice el científico. "Si tomamos esta regla como válida —agrega—, si suponemos que la Naturaleza no da vueltas innecesarias en la evolución, podemos armar programas que busquen cuál es el camino más corto entre dos especies".

Nunca la naturaleza va a hacer que una especie vegetal que tenía una hoja de 3 centímetros de largo evolucione hasta tener una de 5 cm pasando por tener una de 30 cm de longitud. Va a ir de los 3 cm a los 5 cm por el camino más corto: 3,5; 4; 4,5 cm. Si tomamos un conjunto de caracteres de determinadas especies (por ejemplo el color de la flor, el tamaño y la forma de las hojas, el tipo de raíz, etc.) y los cargamos en la máquina, ésta podrá establecer los grados de relación o parentezco que existe entre unas y otras. Se llega así a dibujar el "árbol genealógico" de una planta, descubriéndose qué especie proviene de la evolución de qué especie la historia de la evolución empieza a abrir sus puertas a la informática.

Claro que no es la única aplica-



ción que tiene la computadora para un estudioso de la naturaleza. Las enormes colecciones que existen en el Museo de La Plata podrán ser organizadas y clasificadas con ayuda de las computadoras: ya se encuentra en marcha un estudio piloto relacionado con los materiales tipo de las especies de hongos del Instituto Spegazzini del Museo (unos 4.500 ejemplares).

Con las máquinas también podrán realizar un banco de datos especializado en las ciencias naturales. Y podrán agilizar la publicación de los trabajos científicos realizados por los investigadores de ese centro de estudios y tal vez algún día, también, accedan a los programas (que ya existen en el exterior) que permiten reconstruir cómo era el organismo de un animal a partir de un fósil.

Pero el Museo no es sólo un museo: en él funciona, además, la Facultad de Ciencias Naturales de la Universidad de La Plata. En el mismo edificio donde duermen su sueño de siglos los dinosaurios y pterodactylus, los alumnos de la facultad asisten a sus clases. Y allí también se pueden aprovechar las computadoras.

"Estamos realizando una experiencia en educación a la que hemos denominado CAI", dice el doctor Crisci. El CAI es un sistema de enseñanza personalizada con asistencia de una computadora en el cual la máquina lleva un registro del desenvolvimiento del alumno. El aprendizaje se realiza a través de la interacción entre el alumno y la máquina: esta presenta al alumno un texto que debe leer y luego le realiza una batería de preguntas sobre el tema.

"La dificultad consiste en poder plantearle a la máquina todas las posibles respuestas a las preguntas, aún las erradas —dice Crisci—, para que así pueda reencauzar al alumno hacia el camino correcto cuando se equivoca".

La computación en el aula de Ciencias Naturales permitirá simular fenómenos naturales para su aprendizaje, colaborar en la presentación y revisión de temas, disminuir el aburrimiento asociado con la ejercitación más simple y seguramente mucho más con el tiempo irán descubriendo nuevas facetas.

El equipo con que cuenta el Museo está compuesto por 3 IBM PC, con monitor para gráficos, dos unidades de diskettes e impresora, que se pueden conectar al computador central: una IBM 4331.

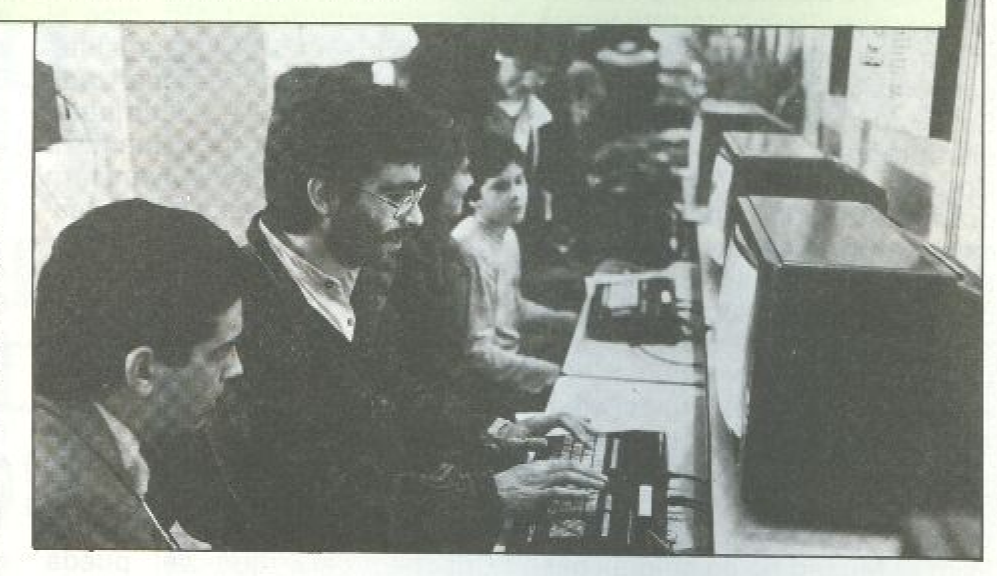
El equipo humano (es lo más importante) está compuesto, por su parte, por 2 antropólogos, 3 zoólogos, 1 ingeniero en sistemas, 2 palenteólogos, 2 ecólogos, 2 botánicos y 2 estudiantes de informática.

Todos ellos, en forma conjunta y con el apoyo de esa batería informática, se encuentran investigando día tras día. Quizás mañana surja en una pantalla el "eslabón perdido" de nuestra raza.

BEGGINERS

IDIOMAS DE COMPUTACION

Para los que recién se inician continuamos brindándoles un resumen de los lenguajes de programación más populares, de manera que puedan realizar su opción.



APL (A Programming Language)

APL fue creado en IBM por Kenneth Iverson. Siempre fue considerado un lenguaje interno de la empresa, hasta que en el año 1960 fue lanzado como lenguaje interactivo para ser utilizado en computadoras grandes. APL fue originalmente un lenguaje científico, con una llamativa facilidad para crear y manipular matrices multidimensionales.

Finalmente, algunos trabajadores de IBM dejaron la compañía y comenzaron su propio desarrollo de sistemas en base a este nuevo lenguaje. Estos servicios extendieron las características de APL en el área de manejo de archivos, para convertirlo así en un poderoso lenguaje con orientación comercial.

Los mayores devotos del APL en la actualidad son las compañías de seguros y también las aerolíneas. Esto es debido a sus rutinas realmente complejas, y su tipo especial de documentación.

Las primeras versiones de este lenguaje aptas para ser usadas en microcomputadoras, fueron hechas para el sistema operativo CP/M. Sin embargo, desde entonces la popularidad de este lenguaje no ha crecido entre los usuarios de este tipo de máquinas. Las razones son varias. Primero, tiene un set de caracteres que es distinto de cualquier otro. Por ejemplo, algunos de los caracteres requieren de dos teclas distintas, combinadas con una de retroceso en el medio, y esto no sólo es incómodo sino también inadaptable para algunas terminales. No utiliza el código ASCII, sino otro llamado código Z, totalmente distinto al standard. Además, todas las operaciones en APL son evaluadas de derecha a izquierda. En nuestra cultura, estamos acostumbrados a leer de izquierda a derecha, y esto puede ser un poco confuso.

Por otra parte, APL utiliza operadores muy complejos que permiten al programador expresar ideas complicadas en una sola línea. Esto tiene la ventaja de consumir poca memoria, pero los programas terminan siendo jeroglíficos incomprensibles una vez que olvidamos el propósito de los mismos.

Por otra parte, la velocidad de ejecución es rápida a la hora de hacer cuentas, pero cuando se trata de loops o decisiones esta velocidad desaparece y nos preguntamos para qué tanto lío, porqué no tomamos el BASIC y listo.

Se trata por otra parte de un lenguaje interpretado, y si bien se siguen viendo algunas nuevas versiones especialmente para las PC's, si comparamos su número con las de BASIC, C o Pascal, veremos que no es lo que se dice un lenguaje popular.

Pascal

Aproximadamente al mismo tiempo en que el FORTRAN era desarrollado en los EE.UU., otro lenguaje era creado con el fin de resolver algoritmos complejos. Su nombre era ALGOL (ALgorithmic Language) y se tornó muy popular en Europa. Se trata de un excelente lenguaje, aunque algo difícil de aprender. En 1971, Niklaus Wirth, de Suiza, inventó el Pascal como una herramienta para la enseñanza del ALGOL y un medio de demostrar los principios de la programación estructurada.

El Pascal es un idioma sencillo de aprender, y aunque un poco más rígido que el BASIC, esta rigidez está plagada de buenas costumbres. Definir variables antes de usarlas y la no existencia del GOTO son dos cosas comunes en la vida de los programadores de Pascal. El lenguaje se ha tornado muy popular, especialmente desde que se comenzaron a ver versiones del mismo para las pequeñas computadoras.

Igual que el BASIC, Pascal "habla" en inglés, y utiliza los signos matemáticos convencionales. Puede realizar operaciones matemáticas complejas, y manejar cadenas de datos y caracteres.

Los tipos de datos con que trabaja Pascal son los comunes, como ser reales, enteros o lógicos, pero permite además que el usuario defina nuevos tipos de datos, del mismo modo que nuevos procedimientos y funciones.

Pascal es un lenguaje compilado, pero generalmente no se compi-



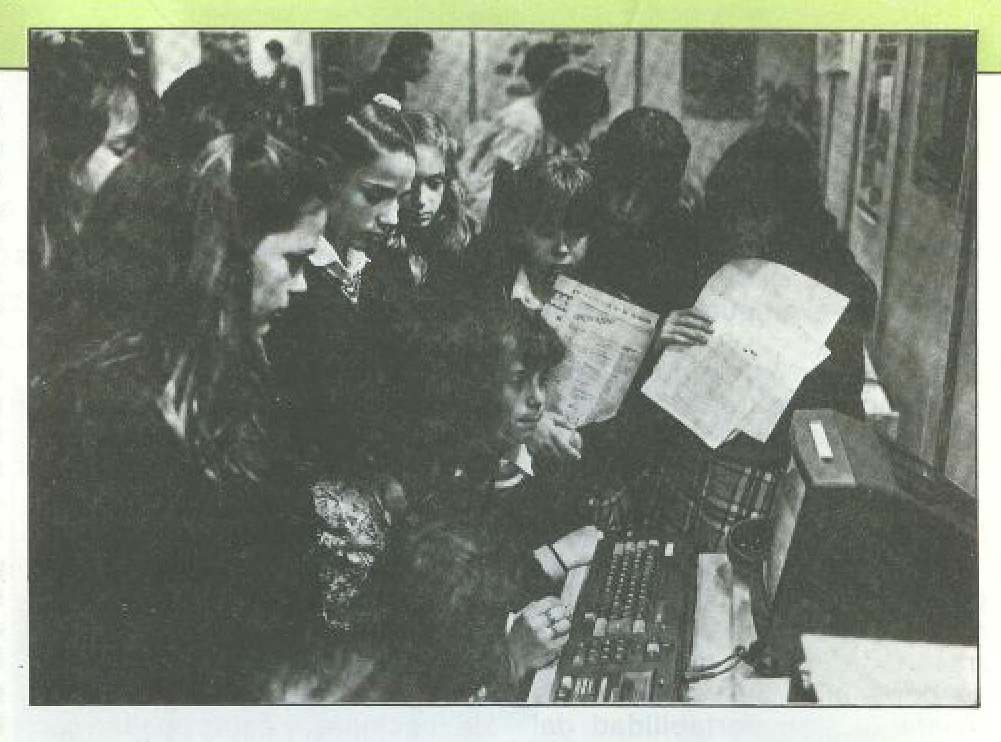
la en código de máquina. Se suele utilizar un código intermedio, llamado **pseudo-code** o simplemente p-code. Este código es guardado como archivo, y en el momento de ejecutarlo es interpretado en el lenguaje natural del micro.
La ventaja de este método, es la de crear un programa completamente transportable, dado que lo único que habría que cambiar sería el intérprete de p-code.
Sin embargo, algunos fabricantes prefieren compilar sus programas directamente en el código del microprocesador que corresponda. De este modo, si bien se gana algo de velocidad, se pierde la transportabilidad del mismo.

LENGUAJE C

Fue creado en los laboratorios Bell, y designado primariamente para operar bajo el sistema operativo UNIX.
Es un lenguaje estructurado, con algunos parecidos con el Pascal. Sin embargo, mientras que Pascal utiliza tanto funciones como procedimientos, C mantiene su modularidad tan solo en base al uso de funciones. Toda la estructura del programa se arma mediante el uso de funciones, llegando al extremo de que no existan sentencias como PRINT o READ.
Las entradas y salidas se llevan a cabo por medio de funciones. Existen los típicos "if-then-else", "while loops", variables locales y globales, matrices y cadenas.
C es un lenguaje compilado, en el que los programas son hechos por medio de un editor de textos, y luego compilados en versiones en código máquina listas para ser corridas en la computadora.
Si bien no utiliza ningún tipo de estructura de entrada/salida propia, ésta es tomada del sistema operativo correspondiente.
Un programa en C es un conjunto de funciones. La habilidad del programador para crear el juego de funciones que necesite para su trabajo, hacen del C un lenguaje altamente flexible. En C no existen los números de línea. El programa empieza con el nombre de una función, y luego un corchete para indicar la definición de la función. Existen bibliotecas de funciones, y éstas pueden ser llamadas desde otra función. También podemos llamar a rutinas en lenguaje de máquina.
Luego de la creación de C para el sistema operativo UNIX, fue transportado a CP/M. De ahí, su paso al MS-DOS fue directo, y actualmente es uno de los lenguajes con mayor futuro en el área de la programación estructurada.

ADA

El lenguaje COBOL se transformó en un standard comercial debido al pedido hecho por las agencias del gobierno, allá por 1960. Hace un tiempo, el departamento de Defensa de los Estados Unidos decidió que un nuevo lenguaje debía coordinar las necesidades de aplicación de sus tres fuerzas, la Armada, la Marina y la Fuerza Aérea. El proceso de creación de tal lenguaje comenzó allá por 1975. Al mismo tiempo, un estudio intensivo de los lenguajes existentes hasta el momento debía averiguar si existía un lenguaje que cumpliera las características de univesalidad necesitadas, y de no existir crearlo de inmediato. Una de las recomendaciones fue que tanto Pascal, ALGOL o PL/1 fueran utilizados como un punto de arranque para la nueva meta. Finalmente fue el Pascal quien formó la base, y el nuevo lenguaje fue creado a partir de él.
Llamado en un principio el lenguaje "verde", luego se cambió su nombre por el de Ada, en honor a la primera programadora, Lady Ada Augusta Byron, hija del poeta Lord Byron.
Desde entonces, Ada ha venido soportando cantidades de tests y pruebas, y varios compiladores ya fueron escritos para el mismo.
Ada es bastante similar al Pascal. Tiene una parte de declaraciones, y otra de sentencias. Es muy rígido en cuanto a su formato, todos los identificadores deben ser declarados y sus atributos especificados.
Las estructuras de control más importantes son dos: las sentencias condicionales (seleccionan acciones alternativas), y las sentencias de loops (que especifican repetición de una acción). Ada utiliza distintos tipos de funciones y subprogramas, que son llamados procedimientos. Además, Ada admite dos tipos de módulos, Paquetes y Operaciones.
Los paquetes son utilizados para definir en forma lógica una serie de recursos utilizados en computación.
Operaciones, son trabajos sepa-



rados que se realizan al mismo tiempo en sistemas de tiempo compartido, o grandes computadoras. El término colectivo de esto es "Multitasking" u operación simultánea. Ada es por lo tanto, un lenguaje creado para trabajar con este tipo de sistemas Multitasking.
Si bien se pueden hacer cosas similares en Pascal o C, Ada es el primer lenguaje creado para aprovechar las características del procesamiento múltiple, y cantidades casi ilimitadas de memoria.

FORTH

Este es otro lenguaje creado por un solo hombre, Charles More, y su primer uso fue el control del telescopio en el observatorio de Kitts Peak. Fue luego extendido por un grupo de programadores que lo hicieron conocer, llevándolo al alcance de las microcomputadoras .
Forth no es un lenguaje fácil de aprender, dado que es un tanto distinto de lo que estamos acostumbrados a ver. Además, todos los cálculos se realizan por medio del sistema de Notación Polaca Inversa (RPN), el mismo que utilizan las calculadoras Hewlett-Packard. Es llamado un lenguaje inacabado, porque el programador tiene la posibilidad de crear nuevas palabras dentro del mismo. En Forth, todo es una palabra (que es otra forma de llamar a una función). No es muy apropiado para manejar grandes cuentas matemáticas, pero puede llamar a subrutinas, perdón, a palabras, escritas en otro lenguaje más apto para este tipo de asuntos. El programador puede definir nuevas palabras utilizando otras ya creadas, y casi sin darse cuenta, el trabajo está hecho. Además, una vez terminado un programa, que en definitiva será una palabra, éste estará en la máquina cada vez que la prendamos, y podremos utilizar todas las palabras que ya hayamos definido en él como si fueran parte del lenguaje.
¿Ven ahora por qué es un poco difícil de entender?
Cuando uno ve por primera vez un programa escrito en Forth, es un poco confuso porque todo parece correr al mismo tiempo, pero luego las cosas comienzan a tener sentido.
En Forth, todas las operaciones se hacen por medio del stack. Este es algo así como una pila donde se almacena información. Si bien todos los lenguajes utilizan un stack en forma interna, en Forth este es directamente accesible por parte del usuario.
Todavía existen algunas cosas normales en Forth. Es un lenguaje estructurado, sin etiquetas ni GOTOs. Es un lenguaje interpretado, que luego se compila y por lo tanto necesita muy poco espacio en la memoria de la máquina. Un idioma Forth entra en tan poco como 16 K de memoria, por lo que no es difícil conseguir versiones del mismo para casi todas las microcomputadoras.
En definitiva, se trata de un lenguaje interesante, aunque si queremos especializarnos en él le tendremos que dedicar muchas horas de aprendizaje.

LISP (List Processing language)

Este lenguaje especialmente dirigido a la inteligencia artificial, fue basado en los trabajos de John McCarthy acerca de lenguajes de programación no numéricos, en el año 1960.
El lenguaje LISP fue implementado en el MIT, y desde entonces, en todo tipo de computadoras, grandes y pequeñas.

LISP es un lenguaje no matemático, compuesto por palabras, como cualquier lenguaje. En LISP, tenemos dos tipos de palabras, átomos y listas. Los átomos son las entidades básicas del lenguaje. Específicamente, cualquier combinación de los caracteres del alfabeto, con cualquiera de los diez dígitos decimales (0 al 9) es un átomo, siempre y cuando comience por una letra. Una lista es un conjunto de átomos y de otras listas.
Una lista consiste de un paréntesis izquierdo, seguido por un número de átomos y listas, y un paréntesis derecho.
El lenguaje tiene funciones, variables, y operadores aritméticos, pero tiene una apariencia un poco extraña para programadores BASIC, dado que todas las operaciones se llevan a cabo en RPN.
Una sentencia LISP se parece a una lista, pero encierra un significado y es una forma elemental de programación.
El lenguaje tiene muchas funciones incorporadas, y deja en libertad al programador para crear las propias.
El listado de un programa escrito en LISP suele verse bastante raro, pero para evitar esto, se incluye con el mismo un utilitario que divide el programa en subsecciones, facilitando así su entendimiento.

Se trata de un lenguaje poco aplicado, tal vez porque no es muy conveniente en caso de tener que manejar problemas matemáticos complejos, pero es el número 1 en lo que a inteligencia artificial se refiere.

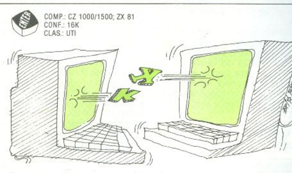
PILOT (Programmed Inquiry Lerning Or Teaching)

PILOT fue el primer lenguaje creado para la enseñanza asistida por computadora. Fue desarrollado en la universidad de California, en San Diego, Estados Unidos.
Es un lenguaje interactivo, y es implementado en computadoras grandes y pequeñas.
El mismo le permite a una persona sin experiencia en programación, crear un programa de diálogo para la enseñanza de alguna materia específica a alumnos que tampoco sepan manejar computadoras.
Usando PILOT, un profesor puede presentar al alumno un pasaje que deba estudiar, para luego interrogarlo mediante un test tipo multiple choice. El programa permite analizar la respuesta del alumno, de modo tal que puede darle mensajes que varían de acuerdo al grado de corrección de éstas.
Se puede también presentar problemas de tipo matemático, y pedirle al alumno que demuestre el camino para la resolución del mismo.
Las instrucciones de PILOT pueden dividirse en varias categorías:
1. Instrucciones de núcleo: estas funciones básicas están formadas por una sola letra, y son comunes a todas las versiones de PILOT. De esta forma, se logra que los programas sean transportables de una máquina a otra.
Estas instrucciones son:
T: TYPE
A: ACCEPT
M: MATCH
J: JUMP
U: USE
E: END
C: COMPUTE
R: REMARK
Existen también algunas instrucciones multipalabra, que no están implementadas en todas las versiones del lenguaje.
2. Instrucciones de cursor y video: se utilizan para determinar en qué zona de la pantalla aparecerá el texto.
3. Instrucciones que determinan distintos parámetros, tales como ports de entrada y salida, velocidad de display, etcétera.
4. Instrucciones de archivos: dedicadas exclusivamente al manejo de datos, y medios de almacenamiento de los mismos.
Como comentario final, podemos decir que se trata de un lenguaje de aplicación específica, el área de enseñanza. Tal vez su mayor problema sea que, desde la popularización del BASIC, y su capacidad para hacer todo lo que PILOT hace, este último ha caído prácticamente en el olvido■

DECODIFICACION DE REM PARA EL CZ 1000/1500

Este soft nos da la posibilidad de leer una sentencia REM y poder ver el programa en código de máquina que éste contiene.

COMP.: CZ 1000/1500; ZX 81
CONF.: 16K
CLAS.: UTI



Tratar con un programa en código máquina para el CZ 1000/1500, es una operación más bien delicada. Esto se debe a que debemos introducir los códigos byte a byte, sin la ayuda de un programa ensamblador. El hecho de que no podamos usar un ensamblador se debe, principalmente, a la limitada capacidad de memoria. Por ejemplo, en el caso de tener 2K de RAM, éste no sería suficiente espacio para dar cabida a un programa ensamblador y además dejarnos espacio para escribir nuestro propio programa.

Por tal motivo, se suele apelar a la introducción de programas en código máquina en una línea REM. La ventaja de este método reside en la simplicidad para luego salvar dicho código, dado que se lo trata como si fuese un programa en Basic como cualquier otro.

Pero, sin embargo, no deja de tener sus desventajas. Por ejemplo, al ir tecleando byte a byte lo que deseamos, tenemos una gran posibilidad de equivocarnos con resultados generalmente desastrosos. Por otra parte, no tenemos posibilidad de verificar lo tecleado pues si deseamos ver el código en una sentencia REM nos encontraremos con una serie de funciones Basic, signos de interrogación y caracteres inidentificables.

El programa que les ofrecemos a continuación nos da la posibilidad de leer una sentencia REM, esto es decodificarla y poder ver el programa en código máquina que ésta contiene. La más obvia aplicación de esto sería el caso de haber tecleado un largo código máquina y querer verificarlo antes de hacer un RUN y obtener algún resultado inesperado o desastroso.

Este programa nos ofrece una presentación de 8 columnas numeradas y 16 instrucciones por columna, lo que hace un total de 128 bytes por página. Se han incluido indicadores especiales para los códigos "CB", "DD", "ED" y "FD" para simplificar la lectura de programas en código máquina. También se añadió otro caracter para señalar el código hexadecimal "76" (118 en decimal) que nos indica la instrucción HALT en Assembler y equivale a ENTER según el set de caracteres del CZ 1000/1500, siendo éste entonces el que señala la finalización de la rutina.

```
7000 PRINT "DECODIFICACION DE RE
M , INGRESE DIRECCION EN DECIMAL
/NUMERO DE  BYTES"
7005 LET H=16
7010 INPUT A
7015 INPUT N
7020 FOR M=0 FO N-1
7025 LET F=INT (M/H)
7030 LET G=F*H
7035 PRINT AT M-G+4,G/4;CHR$ (28
+INT (B/H));CHR$ (28+B-H*INT (B/
H));
7050 IF B=118 THEN PRINT "*"
7055 IF B=203 OR B=221 OR B=237
OR B=253 THEN PRINT "%
7060 NEXT M
7065 PRINT AT 21,0;"DIRECCION DE
 INICIO ";A;" FIN ";A+M
```

El programa

El programa tiene sólo 15 líneas siendo necesariamente corto para poder ser agregado en la parte alta de la memoria que contiene a los programas en BASIC. De ahí que la numeración del mismo comience en la línea 7000.

Las líneas 7000 a 7010 son de presentación e introducción de datos, y el corazón del programa se halla en las líneas 7020 a 7060:

7020: punto de partida del bucle de 0 a N (en realidad de 0 a N-1 pues comenzamos de 0)

7025: calcula el valor de F para numerar las ocho columnas (de 0 a 7)

7030 multiplica F por 16 para darle a G un valor que va 16 en 16

7035: imprime en la línea 2 de la pantalla los números de las columnas (F+1 para obtener de 1 a 8 y tabulación automática por G/4)

7040: le da a B el valor contenido en la dirección A+M

7045: traduce e imprime a 2 caracteres el código hexadecimal de cada dirección

7050: imprime el caracter *

7055: imprime los apuntadores para las instrucciones CB, DD, ED y FD

7060: incrementa la dirección

7065: imprime las direcciones y finaliza K64

diálogo - D.P.

Títulos no oficiales

ROBOT COMANDADO POR LA COMMODORE 64

Diseñado y construido por el equipo que dirige el ingeniero Enrique Distéfano, está pensado especialmente para la enseñanza y el entretenimiento, pero es el primero de una serie que abarcará ejemplares de mayor tamaño y potencia.



Ing Distéfano

Tatú 1°

Mide apenas treinta centímetros de altura y nació hace pocos meses, pero ya es una personalidad: es Tatú 1°, un robot de tecnología totalmente nacional.

"Es un robot eléctrico de tipo brazo articulado", dice el ingeniero Distéfano, quien agrega que tiene "brazo, antebrazo y una mano con dos dedos".

Una vez que se superó la imagen de ficción del robot con forma de androide, se empezaron a fabricar, en todo el mundo, robots de uso principalmente industrial que imitan el brazo humano. Son máquinas capaces de trasladar objetos en el espacio con cierto grado de precisión y que pueden ser programados para realizar determinadas tareas que, por monótonas o riesgosas, el hombre no quiere hacer. Tatú es un integrante de esa familia.

Su figura poco atractiva, y la simpleza de sus líneas no hacen sospechar las posibilidades de este pequeño robot. "Tiene cuatro grados de libertad", dice Distéfano. Eso significa que es capaz de trasladar un objeto desde cualquier punto hasta otro cualquiera, siempre que se encuentren dentro de su radio de alcance.

Los grados de libertad de los robots son los que definen su autonomía para realizar las tareas que se les encomiendan, y se refieren a la capacidad de moverse en el espacio. Tatú, por ejemplo, puede realizar giros alrededor del hombro, rotar el brazo y el antebrazo y mover la mano en forma de cabeceo. Cada uno de esos posibles movimientos es un grado de libertad.

Aunque no se lo considera otro "grado" más, Tatú tiene dos dedos que puede apretar en forma de pinza. "Más adelante pensamos hacer otro modelo con otro movimiento más en la mano —dice Distéfano—, pero no incluimos esa posibilidad en este para no encarecerlo: no vale la pena agregar demasiados movimientos a un robot que se usará principalmente en la enseñanza".

Tiene un alcance de 50 cm, con un radio de giro de casi 360 grados (siempre queda un cono o zona de protección, donde el robot no alcanza con su brazo, para seguridad del operador) y en el movimiento arriba-abajo puede recorrer 120 grados. Llega a sostener piezas de hasta 1 kilogramo y a hacer una fuerza, apretando sus dedos, de entre 2 y 3 kilogramos.

Tatú es esencialmente un brazo, pero tiene una "cabeza" inteligente que es la que ordena sus movimientos: una Commodore 64. "Usamos esta computadora por la facilidad que nos representa su port (puerta) de salida", comenta Distéfano. "'En realidad, —dice— el robot puede usarse con cualquier computadora que tenga esa posibilidad de conexión".

El robot es comandado por un programa específico de funcionamiento que incluye todas las operaciones que se deseen realizar. Para indicarle cómo debe moverse se le dan, a través de la computadora, órdenes específicas a los 5 motores eléctricos que mueven las partes de su anatomía.

"Tatú tiene 5 motores eléctricos: uno por cada grado de libertad o tipo de movimiento y otro para el apriete de la mano", dice Distéfano. La exactitud en el movimiento de estos motores es lo que da la precisión del desplazamiento: por ello se deben dar ór-

denes exactas de cuánto tiene que girar cada motor y en qué sentido desplazar cada una de las partes del robot. En el caso de Tatú, tiene una precisión de un milímetro, o sea que del lugar que se le indica adonde debe llegar, puede desviarse en más o menos esa distancia.
Las órdenes a los motores se dan por medio del teclado de la computadora. Para indicarle que se mueva se aprieta la tecla que indica el sentido de giro de cada uno de los motores y el número de pasos (la cantidad de movimiento) que debe dar.
Tatú fue presentado en sociedad en la exposición EMAC de mayo de este año, pero el grupo de investigadores que lo desarrolló venía trabajando en este proyecto desde setiembre de 1985.
"Ya hemos desarrollado todo el software específico para el funcionamiento, aquello que sería el equivalente del programa residente de una computadora, el programa que le permite realizar las operaciones, trasladar cosas, etcétera". Pero aún se está desarrollando, en colaboración con el Instituto de Informática de la Universidad Nacional de San Juan, el curso que va a acompañar a Tatú: como es un robot dedicado principalmente a la enseñanza, debe tener un programa de entrenamiento para aprender a manejarlo.
Este curso incluye nociones básicas de robótica y ejercicios específicos de manejo. Entre estos últimos puede citarse operaciones tales como tomar un objeto y trasladarlo hasta otro lugar, que es el ejercicio más sencillo; o tomar una pieza y colocarla en una cinta transportadora, o recibirla en la salida de ésta. También se puede hacer que el robot coloque un tornillo, cierre o abra una llave o un interruptor, o incluso que apriete una tecla de una computadora.
A través de este tipo de ejercicios es que se aprende a manejar el robot.
Su campo de aplicación son las escuelas técnicas, universidades y fábricas.
Tatú, según su "padre", tiene un costo acorde a la capacidad adquisitiva del mercado latinoamericano, que es al que piensa dirigirse para exportarlo". El precio de este robot es, aproximadamente, un quinto del de otro similar de fabricación norteamericana (tiene un desarrollo tecnológico anterior, pero es acompañado de un curso más completo).
El bajo costo permitiría usar este tipo de robots en pequeñas industrias, ya que es muy común que se desplacen cargas menores a un kilogramo, algo que Tatú puede hacer.
El lenguaje utilizado en la programación de la Commodore 64 que comanda el robot es el Basic, "pero las rutinas de control de los motores están dadas en lenguaje de máquina —aclara el ingeniero—, ya que éste se adapta mejor a la tarea".
Pese a lo que se puede suponer lo más difícil fue el desarrollo de la parte mecánica: los motores de corriente continua que se utilizan, las transmisiones, el censado de los pasos de los motores, recuerda Distéfano. En cambio, el desarrollo del software necesario no fue tan complejo.
Existen dos tipos de robots: los eléctricos y los neumáticos. Los más difundidos son los primeros, ya que son más precisos y tienen mayor potencia (o sea, la capacidad de levantar mayores pesos a mayores distancias). Tatú, decíamos, es un robot eléctrico, pero de pequeño tamaño.
Un robot neumático no es capaz de levantar más de 10 kilogramos con un brazo de más de un metro de longitud. Pero la ventaja de éstos es que por su mayor sencillez (en lugar de un motor por cada movimiento tienen uno sólo para el compresor) son mucho más baratos que los eléctricos.
Distéfano (quien también es doctor en física) dirige el grupo de investigadores que se encuentra abocado, también, al desarrollo de robots neumáticos, así como el de eléctricos de mayor potencia. En el proyecto de fabricación de los primeros cuentan con el apoyo financiero del Area Sábato del Banco de la Provincia de Buenos Aires, la que se especializa en el apoyo al desarrollo de nuevas tecnologías.
Tatú, con sus 30 centímetros de altura, su brazo mecánico y sus cinco motores, se ha convertido en un logro de la tecnología nacional y también en el principio de una importante industria. Pero, por sobre todo, se ha convertido en otra posibilidad para los usuarios de la Commodore 64 ■

Marcelo Cantón

INTERFASE UNIVERSAL
PARA CONECTAR CUALQUIER COSA

Si queremos que la computadora controle dispositivos para que, por ejemplo, algo se prenda o se apague, se mueva o se quede quieto, este proyecto es especialmente adecuado. Está hecho para ser usado con CZ 1000/1500, Spectrum, TK 83/85 y TK90X.

En este artículo, veremos cómo construir la interfase para las distintas computadoras, y en una próxima nota les daremos algunas aplicaciones prácticas para la misma.

El circuito fue creado para poder conectar hasta 12 periféricos a una CZ 1000 o una Spectrum, sin que sea necesario modificar el mismo. Si bien no es necesario tener grandes conocimientos de electrónica para construirlo, supondremos que el constructor ya ha tenido alguna experiencia manejando un soldador.

Esta interfase nos permitirá conectar a nuestra computadora una gran variedad de periféricos, como ser ports serie y paralelo, conversores analógico-digital y digital-analógico, impresoras, joysticks, etcétera.

Con el objetivo de lograr una mayor versatilidad, la interfase se compondrá de una plaqueta de circuito impreso (o alguna otra cosa) en la que estarán montados todos los componentes de la misma. Estos pueden dividirse en dos grandes grupos, el primero serán los circuitos integrados y componentes necesarios para el funcionamiento de la interfase en sí. El segundo, será una serie de zócalos para circuitos integrados que serán dejados libres (es decir, sin ningún integrado colocado en ellos). Estos zócalos serán de 14 pines cada uno y se utilizarán como conectores. En ellos se conectarán los distintos periféricos que utilicemos con esta interfase.



En cada zócalo se podrán conectar hasta tres periféricos independientes, o dos, en caso de que sean necesarias las líneas de control.

A cada zócalo, llegarán las siguientes señales: las líneas de datos D0 a D7, READ, WRITE, VCC, GND, y dos líneas de habilitación (PE).

La interfase le asigna tres números de periférico a cada zócalo. En la tabla 1 podemos ver las asignaciones, tanto en decimal como en hexadecimal.

Como señalamos antes, cada zócalo tiene asignadas dos líneas de habilitación. Estas pueden tomar los valores 0 o 1. Cuando ambas líneas toman el valor 1, el periférico es deshabilitado.

Tabla 1

ZOCALO	PE	NRO. PERIFERICO	HEXA
1	00	252	FC
1	01	253	FD
1	10	254	FE
2	00	243	F3
2	01	247	F7
2	10	251	FB
3	00	63	3F
3	01	127	7F
3	10	191	BF
4	00	207	CF
4	01	223	DF
4	10	239	EF
no sel.	11	255	FF

Tabla 2

DIRECCION	DATO
16514	62
16515	00
16516	211
16517	127
16518	201
16519	62
16520	00
16521	211
16522	191
16523	201
16524	6
16525	0
16526	219
16527	191
16528	79
16529	201

Tabla 3

TERMINAL	FUNCION
1A	DATA 7
4A	DATA 0
5A	DATA 1
6A	DATA 2
7A	DATA 6
8A	DATA 5
9A	DATA 3
10A	DATA 4
15A	IORQ
16A	RD
17A	WR
4B	MASA
5B	MASA
19B	A7
20B	A6

SELECCIÓN DEL PERIFÉRICO

Cada periférico tendrá asignado un número correspondiente, como lo podemos ver en la tabla 1. El número correspondiente al periférico que querramos direccionar debe ser pokeado en la dirección 65407. Los datos que querramos leer o escribir deberán ser pokeados o leídos en la dirección 65471.

Si estamos trabajando con una Spectrum, esto puede ser hecho desde Basic. Por ejemplo, si hacemos:

OUT 65407,254

estaremos seleccionando el periférico número 254

LET A = IN 65471

leerá los datos presentes en el port seleccionado (en este caso el 254) y los guardará en la variable A.

Si cambiamos esta última instrucción por:

OUT 65471,A

hará que el contenido de la variable A pase al periférico previamente seleccionado.

La programación de la CZ 1000 es un poco más complicada, dado que las instrucciones del BASIC no incluyen comandos directos de entrada/salida. En la tabla 2 podemos ver tres cortas rutinas en código máquina que se encargarán de manejar las entradas y salidas de y a los periféricos. Esta rutina está almacenada en una línea 1 REM, en la cual se deben dejar 16 bytes en blanco. Estos serán ocupados luego por la rutina antes mencionada.

Para seleccionar un periférico en la CZ 1000, debemos hacer lo siguiente:

POKE 16515,N Donde N representa al número de periférico seleccionado

RAND USR 16514 Y el periférico es seleccionado.

Para mandar datos al periférico seleccionado, hacemos:

POKE 16520,D. Donde D es el dato a enviar o el nombre de una variable que lo contenga.

RAND USR 16519 El dato pokeado en la dirección 16520 es enviado al periférico seleccionado.

Para hacer un ingreso de datos, hacemos: LET A = USR 16524

El dato es ingresado desde el periférico y almacenado en la variable A.

CÓMO FUNCIONA

En la figura 2, podemos ver un esquema eléctrico de la interfase propuesta. Los diodos D1 y D2, junto con la resistencia R1 forman una compuerta OR, cuya salida será cero sólo cuando se intente escribir al port 65407. Estos componentes son necesarios, ya que la línea IORQ se pondrá en estado lógico cero cada vez que la CPU reconozca una interrupción. Dado que ni RD ni WR se pondrán en cero bajo esta condición, esta compuerta asegura que no habrá interferencia por parte del sistema de interrupciones de la CPU. La compuerta IC1d hace un OR entre esta señal y la línea de address

A7. Esta última es la línea de selección de periférico. Cuando las líneas IORQ, WR, y A7 están bajas, los circuitos integrados IC2 e IC3 "traban" en sus líneas de salida el dato presente en las líneas de entrada. Entonces, cada vez que se dé la situación antes mencionada (IORQ, WR y A7 bajas), trabaremos en la salida el dato que enviamos al periférico.
La línea A6, es utilizada para manejar la entrada y salida de datos. Junto con las líneas IORQ, RD o WR, generará a la salida de ICia e ICib las señales lectura de port (PRD) y escritura de port (PWR). Estas dos líneas estarán siempre en 1, excepto cuando se efectúa una lectura o escritura al periférico.
Con toda esta teoría, una interfase se paralela es fácilmente realizable bajo estas condiciones. Para escribir un dato en un periférico, lo único necesario es hacer un OR entre la línea PE (habilitación de port) y la línea PWR. Con esta señal, podemos avisarle al periférico que el dato está listo para que lo reciba. Para realizar la entrada de un dato, el procedimiento es el mismo. Se hace un OR entre la línea de PE que se desea y la línea PRD.

CONSTRUCCIÓN

Para llevar a cabo este proyecto, necesitamos un soldador de menos de 50 Watts, para no dañar los componentes electrónicos. Es recomendable utilizar un circuito impreso diseñado ad hoc, aunque una plaqueta universal también se puede adaptar a nuestros fines. La última opción sería hacerlo con el método llamado Wire Wrap, bastante práctico pero fuera del alcance económico de la mayoría. Este consiste en ir uniendo las patitas de los distintos zócalos y componentes por medio de un fino cable aislado. El mismo se enrolla alrededor de los pines, por medio de una herramienta especial que le quita la aislación al cable. De esta forma, se van tendiendo cablecitos de pin a pan, hasta completar el ensamblado.
Para aquéllos que no puedan acceder a este método, el soldador será la única solución.
En la figura 3 podemos ver una disposición ideal de los componentes sobre la plaqueta.
Los cuatro zócalos para la posterior conexión de periféricos están situados en un extremo de la plaqueta para una mayor comodidad. No debemos olvidar que a estos cuatro zócalos van todas las líneas de address, y un cortocircuito entre estas sería fatal.
Es recomendable el uso de zócalos para los tres integrados, IC1, IC2 e IC3.
También debemos tener en cuenta la polaridad de los diodos al soldarlos, ya que de ponerlos al revés, la interfase no funcionará. En cuanto al conector de borde, éste debe tener al menos 23 contactos, y en la tabla 3 podemos ver los pines que utilizaremos en el caso de la Spectrum o la CZ 1000. Estos deben ir conectados al circuito, según se ve en la figura 2 y excepto las líneas de datos que van conectadas a varias terminales simultáneamente. Las restantes están conectadas sólo a los lugares indicados en la figura.

Figura 1

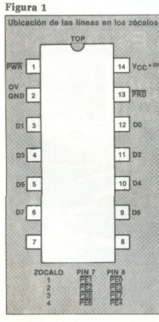
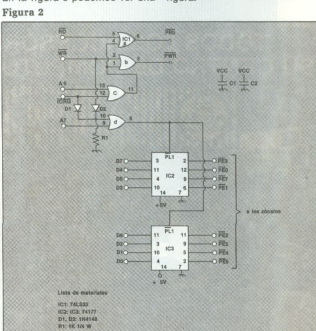


Figura 2



PARA VENCER A LOS TURBOS

Ing. Pedro E. Colla

Sin querer ser "piratas", lo cierto es que es necesario obtener copias de superioridad de los programas para la eventualidad de que un accidente los destruya (para TS 2068 y Spectrum).

La copia de software viene siendo centro de una intensa polémica; no tanto en nuestro país, que carece de **software houses** de importancia y de agrupaciones fuertes de usuarios.
Tanto los primeros como los segundos exponen razones de verdadero peso en contra y a favor del copiado de software.
El análisis detallado de las causas y efectos es un tema lo suficientemente extenso y complejo como para que escape al tratamiento en esta nota. Casi tanto como para, quizás, generar un artículo dedicado al tema en el futuro.
De todas maneras, si bien la copia de software para evitar su compra no sólo es ilegal sino nefasta para la futura disposición de programas frescos, sigue siendo necesario obtener copias de seguridad de los programas que uno posee para la eventualidad de que un accidente los destruya.
Como sea, los programas vienen provistos de técnicas de protección cada vez más sofisticadas.
Lo más reciente en esta materia, y no exclusivo de los computadores Sinclair, es la utilización de TURBOS.
Para entender cómo funcionan estos es necesario ver rápidamente cómo es el proceso de lectura/grabación por cassette.
Cada modelo de computador utiliza el formato de grabación que por diversos motivos le resulta más conveniente.
En general, la carga de programas implica que el usuario exprese mediante las instrucciones apropiadas qué programa desea, invocando de esa manera las instrucciones del sistema operativo encargadas de la grabación o reproducción por cassette según el caso.
Una vez finalizada la carga se retorna el control al usuario o al programa para su ejecución. Esto depende de las opciones utilizadas; en general los programas comerciales hacen esto último.
El problema de copiar un software determinado consiste en que una vez que el mismo resida en la memoria se nos permita de alguna forma detenerlo para de esa manera poder salvar la copia.
Si el programa está en BASIC es ma-

Figura 1

```
L
00010 ;------------------------
00020 ;COPYALL
00030 ;
00040 ;Programa para copia
00050 ;de software sin
00060 ;limitacion de metodo
00070 ;de grabacion
00080 ;------------------------
00090          ORG 55000
00100          ENT
00110 ;------------------------
00120 ;PROGRAMA PRINCIPAL
00130 ;EFECTUA UN RETARDO
00140 ;------------------------
00150 BEGIN    LD BC,#FFFF
00160 WAIT     DEC BC
00170          LD A,B
00180          OR C
00190          JR NZ,WAIT
00200 ;------------------------
00210 ;APAGA INTERRUPCIONES
00220 ;------------------------
00230          DI
00240 ;------------------------
00250 ;MUEVE STACK A ZONA
00260 ;SUPERIOR DE MEMORIA
00270 ;------------------------
00280          LD (STACK),SP
00290          LD SP,56000
00300 ;------------------------
00310 ;RUTINA PRINCIPAL
00320 ;LEE PORT DE CASSETTE
00330 ;------------------------
00340 MAIN     IN A,(#FE)
00350 ;------------------------
00360 ;VERIFICA SI ES break
00370 ;------------------------
00380          RRA
00390          JR NC,BREAK
00400 ;------------------------
00410 ;SELECCIONA BIT 6
00420 ;(ENTRADA ear)
00430 ;COMO SE HIZO UNA
00440 ;ROTACION PREVIAMENTE
00450 ;EL BIT 6 ESTA AHORA
00460 ;REALMENTE EN BIT 5
00470 ;------------------------
00480          AND 32
00490 ;------------------------
00500 ;ROTA A BIT 3 (mic)
00510 ;------------------------
00520          RRA
00530          RRA
00540 ;------------------------
00550 ;EL CONTENIDO DE A
00560 ;TIENE AHORA EL BIT 6
00570 ;LEIDO SOBREEL PORT
00580 ;DE CASSETTE EN BITS
00590 ;4,3 Y 2
00600 ;EMITE POR EL PORT DE
00610 ;CASSETTE
00620 ;------------------------
00630          OUT (#FE),A
00640 ;------------------------
00650 ;SE ROTA SOBRE BIT 2
00660 ;EL QUE ES border DE
00670 ;VIDEO PARA DAR
00680 ;INDICACION VISUAL
00690 ;------------------------
00700          RRA
00710          OUT (#FE),A
00720 ;------------------------
00730 ;REPITE EL CICLO
00740 ;INDEFINIDAMENTE
00750 ;------------------------
00760          JP MAIN
00770 ;------------------------
00780 ;RUTINA DE RETORNO
00790 ;COLOCA EL STACK EN
00800 ;SU ANTERIOR POSICION
00810 ;------------------------
00820 BREAK    LD SP,(STACK)
00830 ;------------------------
00840 ;HABILITA INTERRUP.
00850 ;------------------------
00860          EI
00870 ;------------------------
00880 ;RETORNA A BASIC
00890 ;------------------------
00900          RET
00910 STACK    DEFB #00,#00
00920 END      NOP
A
S
END            = #D701
BREAK          = #D6F9
MAIN           = #D6E8
STACK          = #D6FF
WAIT           = #D6DB
BEGIN          = #D6D8

T
Start of source = 32768
Length          = 01708
```

teria sencilla el detenerlo pues, en algunos casos, bastará oprimir la tecla **break** y en otros realizar la carga mediante **MERGE**; si el programa tiene secciones en lenguaje de máquina bastará con cargar y salvar estas a "mano" y no bajo el control del programa para consumir la copia.
Si en cambio el software está escrito totalmente en código de máquina, una vez que el programa tomó control no habrá forma de detenerlo; en estos casos se utilizan algunos métodos ingeniosos para cargarlos de tal manera que siempre se autoejecuta y por lo tanto es imposible detenerlo.
Para poder copiar software de este tipo habrá que disponer de un copiador como el presentado en el número 4 de esta revista; es decir un programa en lenguaje de máquina que, utilizando las rutinas apropiadas del ROM, cargue en una zona de memoria cualquiera desde el cassette e inmediatamente salve esa zona de memoria.
En este último caso, tanto el programa leído como el salvado utilizan el formato standard del computador.
Para evitar la utilización de este método se crearon los programas TURBO. Consisten en una rutina de lectura y grabación que utilizan un formato distinto al que usa el computador normalmente.
De esa manera algo grabado por una rutina TURBO solamente puede ser leído por su par destinada a la lectura, careciendo de sentido para la instrucción LOAD standard.
Normalmente, el formato standard de grabación consiste en colocar pulsos de audio en un port del computador relacionados con el contenido en bits de la información a salvar y temporizando el ritmo de colocación mediante retardos de software.
Una forma sencilla de hacer un TURBO es mediante la reproducción de las mismas rutinas de LOAD/SAVE del computador, pero alterando las constantes de retardo; de esta manera se aumenta la velocidad de transferencia y, si bien el método de lectura prevee cierta tolerancia a la variación de velocidad, más allá de un límite no puede manejarlo.
En un programa que utilice este mé-

Figura 2. Disposición de Port FEh

Para Lectura (IN)

b7	b6	b5	b4	b3	b2	b1	b0

- b0–b4: TECLADO
- b5: NO USADO
- b6: EAR
- b7: NO USADO

Para Salida (OUT)

b7	b6	b5	b4	b3	b2	b1	b0

- b5–b7: NO USADOS
- b0–b2: BORDE
- b3: MIC
- b4: Parlante BEEP

DESPROTEGIENDO SOFTWARE

todo se procede, en primer término, a la carga de una sección en lenguaje de máquina, llamada BOOT o BOOTER, la que contiene únicamente las rutinas del TURBO; éstas capturan a continuación el resto del programa utilizando el formato "cambiado".
Si bien el BOOT está grabado en forma convencional y protegido por algunos de los métodos ya explicados, el detener su ejecución antes de que tome control o copiar el mismo no tiene ningún sentido, pues el programa en sí todavía no reside en memoria.
Un método para evitar este inconveniente es la copia de grabador a grabador, pero se tropieza con el inconveniente de que sin disponer de dispositivos de ecualización de la señal la misma se degrada mucho y no siempre por este motivo es posible realizar la copia.
Una forma más refinada, siempre mediante dos grabadores, es dejar al computador que se encargue de los niveles de señal con sus propios circuitos al efecto; para realizar tal tarea se requiere un pequeño programa en lenguaje de máquina.
El mismo está expuesto en la Figura 1 y su funcionamiento es muy sencillo.
Al invocarlo se inspecciona el port de entrada de cassette y sea lo que sea que allí se encuentre se lo traslada al port de salida en una sucesión continua; dado que desde el momento en que se efectúa la lectura hasta que se efectúa la grabación pasan algo menos de 30 microsegundos (treinta millonésimas de segundo) la señal en el port MIC será una réplica fiel de la presente en el port EAR.
Como puede observarse el port de cassette, tanto para la entrada como para la salida tiene la dirección FEh (254 decimal) y el contenido de esos ports puede verse en la Figura 2 donde quedan en claro las múltiples funciones de los mismos.
Nótese que antes de comenzar se realizan dos tareas, ambas muy importantes: se apagan las interrupciones y se corre el stack pointer.
El computador cada 16 microsegundos recibe una interrupción enmascarable la cual lo obliga a inspeccionar el teclado y a almacenar las teclas que estuvieron oprimidas; esto provoca que el timming tenga "lagunas" durante el tiempo de cada interrupción; el resultado es totalmente indeseable a los efectos del presente programa y se evita mediante el enmascaramiento o apagado de las interrupciones, mediante la instrucción del Z80 denominada DI (Disable Interrupt).

Para entender la necesidad del corrimiento del stack deberemos brevemente introducirnos en la forma de generación de video.
Tanto en la TS2068 como en Spectrum el video se genera mediante la utilización de un chip especial denominado ULA (Uncomitted Logic Array) y sin la participación del procesador principal Cada procesador tiene su propio bus para operar sin interferirse mutuamente.
No obstante, ambos comparten la utilización de la sección de la memoria RAM que va desde la dirección 16384 a 32767; en esta área el ULA espera encontrar el file de video (en TS2068 pueden ser dos); por su parte el Z80 espera encontrar datos (todas las variables de sistema) y programa BASIC.
Cuando por el azar del procesamiento ambos requieren acceder a esa zona de memoria, y dado que solamente uno lo puede hacer por vez, hay dos soluciones. La primera es dar prioridad al procesador Z80, esto generaría una "chispa" sobre la imagen de video correspondiente al trozo de imagen que no se pudo generar por estar inactivado el ULA.
El segundo método, y el empleado por el Spectrum y TS2068, es que tenga prioridad el ULA. Para llevarlo a cabo cuando este último, que también genera el clock del Z80 como parte de sus múltiples funciones, detecta que el Z80 va a interferirlo simplemente le "apaga" el reloj y lo deja "congelado" durante el tiempo necesario para terminar su tarea; finalizada la cual se "enciende" nuevamente el reloj y el Z80 continúa procesando como si nada hubiese pasado.
Esta interrupción produce una alteración del timing similar a la descripta para la del teclado y es también indeseable.
Para evitarla el programa no debe acceder a rutinas del ROM (las que frecuentemente usan variables de sistema en esa área de memoria) y se coloca el programa en la dirección 55000.

Figura 3. Valores de Borde

B2	B1	B0	COLOR
0	0	0	Negro
0	0	1	Azul
0	1	0	Rojo
0	1	1	Magenta
1	0	0	Verde
1	0	1	Cian
1	1	0	Amarillo
1	1	1	Blanco

En el computador TS2068 el área utilizada para STACK también se encuentra en esa zona (direcciones 24576 a 25088). Por lo que hay que transferirla previo al comienzo de ejecución del programa.
El funcionamiento del resto es sencillo; se lee el port FEh el cual en su bit 6 tiene la entrada EAR, pero vemos también en la Figura 2 que los cinco bits menos significativos de este port son para el manejo del teclado; en especial la tecla **break** hará cuando esté oprimida que el bit 0 tenga un valor 0.
Para detectar esta condición, y por lo tanto poder detener el programa, se rotan todos los bits una posición a la derecha apagando el bit "0" el carry flag en caso de estar oprimida **break**.
Si no está apretada esta tecla, el bit de EAR ahora en la posición del bit 5 debido a la rotación anterior, es aislado del resto de los bits que no representan ulterior interés mediante una operación AND.
El resultado es rotado dos veces más hacia la derecha para que se coloque sobre el bit 3 realizándose a continuación la operación de salida, refiriéndonos a la Figura 2 veremos que el bit 3 es justamente la salida MIC del computador.
Antes de volver a leer se rota una vez más el bit de tal manera que ocupe la posición del bit 2 el cual controla el color del borde (ver Figura 3). De esta forma el mismo se pondrá de color negro cuando el bit ingresante sea "0" y verde cuando sea "1" dando una indicación visual del proceso de reproducción y facilitando la tarea de ajuste de los controles de ambos grabadores de audio.
Por último, antes de retornar a BASIC se habilitan las interrupciones nuevamente (EI) y se restaura el Stack Pointer a su posición original.
El omitir realizar estas tareas conduciría a que el computador quede "colgado" al querer el BASIC leer lo ingresado desde el teclado en una variable de sistema que nunca se actualiza por no haber interrupciones.
Por otra parte el punto al que debe retornar está indicado en el viejo Stack por lo que si no se lo restaura el punto de retorno es espúreo.
Este programa permite copiar **cualquier** software en tanto que el mismo se valga de tonos de audio para registrarse, por lo que si bien su ejecución es válida indistintamente en un computador Spectrum o TS2068, se puede copiar software de otros computadores (TS1000 por ejemplo) ■


¡YA!
CLUB DE USUARIOS
Talent MSX
Cabildo 2027 - 1º (1428) Cap. Fed.
Esmeralda 320 - 5º (1343) Cap. Fed.
Tucumán 2044 - 1º (1050) Cap. Fed.
dialogo - D.P.
A la Talent MSX nada le es imposible
El Club de Usuarios de MSX
ya funciona en sus tres direcciones: CABILDO 2027 - 1º A
ESMERALDA 320 - 5º y TUCUMAN 2044 - 1º - CAPITAL



INTERFASE RTTY

MAQUINA: COMMODORE/2068/ SPECTRUM
FABRICANTE: COMPUTEL

Mediante este periférico, podemos hacer que nuestra computadora trabaje como un eficaz e inteligente receptor y transmisor de señales de radio teletipo (RTTY). Para todos aquellos usuarios de microcomputadoras que estén en el tema de radio, este tipo de transmisión no es nada nuevo. Para aquellos que no estén al tanto, haremos una breve descripción de este sistema de radiocomunicación.

El sistema RTTY fue creado hace varias décadas con el objeto de controlar la escritura a distancia, mediante la emisión de ondas de radio.

Para poder recepcionar una transmisión de RTTY, debemos tener un receptor de onda corta, que trabaje en el modo de banda lateral única.

Aquellos que habitúen escuchar transmisiones de onda corta, habrán escuchado más de una vez sonidos que parecen de código Morse, pero a una velocidad muy elevada. Estas son las transmisiones de RTTY. Sin un decodificador adecuado, estas transmisiones son indescifrables, pero una vez que conectamos la interface a nuestra computadora, podemos ver en pantalla el texto que se esté transmitiendo en ese momento.

Las transmisiones de RTTY cumplen un amplio espectro de informaciones. De ellas podemos recibir noticias de las diferentes agencias nacionales e internacionales (TELAM, ANSA, DYN, AFP, etcétera), y hasta mensajes particulares dirigidos a embajadas. También es un canal de comunicación abierto para los radioaficionados, pudiendo optar por este método de transmisión, tecleando sus mensajes en los teclados de sus computadoras.

En cuanto al modo de operación para recibir las transmisiones, éste es bastante sencillo. Si se trata de transmisiones comerciales, sus distintos parámetros son conocidos (frecuencia de trabajo, velocidad de transmisión, y shift). Una vez alimentada la computadora con estos parámetros, comenzaremos a ver en forma inmediata el texto recepcionado en la pantalla. Este texto puede ser almacenado en un buffer para su impresión posterior.

Si bien no es necesario tener grandes conocimientos de radio para operar el módulo, es necesario entender la terminología básica que se emplea en este medio.

El manual de instrucciones es claro y abarca todas las posibilidades de recepción y transmisión, con ejemplos ilustrativos en cada caso.

El módulo conforma una unidad sellada, donde sólo entra el cable de audífono de la radio, y podemos optar por monitorear la señan ingresada por medio de otra conexión simple.

El mismo se conecta en el port del usuario en el caso de una C-64, o en el conector trasero para una Spectrum o TS 2068.

La alimentación del mismo es obtenida de la misma computadora.

Entre las instrucciones que acompañan al módulo, se incluye una lista con las frecuencias y horarios de las principales agencias noticiosas del país y el exterior. Esta es la mejor manera de enterarse de qué está sucediendo en el país y en el mundo al momento en que la noticia es generada. Es preciso señalar, que esta información no puede ser utilizada en forma comercial, aunque no esté prohibida su recepción en forma privada.

Junto con el módulo se suministra un programa en castellano, que maneja por medio de menúes la operación del sistema. Este programa está desarrollado en código máquina, y siendo fácil de operar, no deja de lado ninguna de las posibilidades del equipo.

Un último modo de operación es el de telegrafía. Este permite transmitir señales sin la necesidad de manipular un interruptor, sólo basta escribir en el teclado de nuestra computadora. Con respecto a la recepción, deberán tomarse los mismos recaudos que para RTTY, considerando además que si el operador que estemos recibiendo está transmitiendo en forma manual, la velocidad de transmisión será variable y podremos obtener errores en la recepción de la misma.

Finalmente, para todos aquellos entusiastas de la radio, les adelantamos el próximo proyecto de Computel. Este consiste en una interface para la recepción de televisión de barrido lento (SSTV), una forma que tienen los radioaficionados de transmitir imágenes de televisión por medio de un transmisor de onda corta ■

CARTUCHO PLANILLA DE CÁLCULO

FABRICANTE:
JDC COMPUTACIÓN
COMPUTADORA:
C-64, C-128 (MODO 64)

JDC Computación
PLANILLA DE CALCULO
CARTRIDGE
para Commodore 64-128
INDUSTRIA ARGENTINA

Este cartridge para C-64 (o 128 en modo 64), nos proporciona una potente herramienta para el manejo de cuadros de valores relacionados entre sí. Se trata de una planilla de cálculo electrónica, cuya novedad consiste en estar totalmente en castellano, tanto sus distintos menúes como instrucciones.

En cuanto al funcionamiento en sí, su carga es automática al encender la computadora. Esta es la ventaja del cartridge frente al disco, es decir nos evitamos el tiempo de carga.

La matriz de esta planilla de cálculo está formada por 15 columnas y 50 filas. Esto nos da un total de 750 espacios disponibles. En cada espacio o celda, podremos almacenar números, quedando descartadas las letras, o títulos de cualquier tipo.

Podemos realizar cualquier tipo de operación matemática entre celdas, filas o columnas.

Entre las opciones que el programa posee, están la de salvar la planilla a disco, cargar otra de disco, listarla por impresora, y ver cuánta memoria libre nos queda para utilizar.

En cuanto a la operación con disco, ésta es un poco lenta debido a que no se puede conectar un acelerador de disco al estar el port ocupado.

El menú general tiene también una opción peligrosa que es la de borrado de planilla. Dado que ésta se ejecuta sin conformación de ningún tipo, debemos estar muy atentos y no apretar esta tecla por error. De todos modos es una buena costumbre guardar periódicamente en disco el contenido de la planilla.

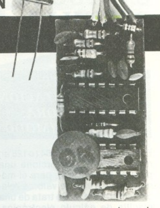
Otra característica a tener en cuenta es que dado que el tamaño de la celda está fijo, el número máximo de dígitos con los que podremos trabajar será 9. Por lo tanto, todo cálculo que implique más de 9 posiciones será redondeado.

Como síntesis, podríamos decir que la principal ventaja es su facilidad de manejo, realizada totalmente por menúes. Esto hace de las instrucciones un complemento, en donde se explica específicamente alguna función.

Su desventaja, a su vez, es que debido a la simplicidad de su uso se han dejado de lado algunas características de planillas de cálculo más avanzadas, aunque no por esto sea menos potente que ellas ■

KIT DE CONVERSIÓN PAL-N

FABRICANTE: HALLEY COMPUTACIÓN
MÁQUINA: TS 2068



Siendo la norma de color de la 2068 la utilizada en Estados Unidos (NTSC), era imposible ver los colores generados por ésta en un televisor que funcione en la Argentina. Ante esta situación, nos restaban dos posibilidades: adaptar la computadora o el televisor.
Para aquellos que no se deciden a mandar a hacer el televisor binorma, Halley pone a su alcance un kit para convertir la TS 2068 a PAL-N.
El mismo consta de una plaqueta y un cristal.
El trabajo a realizar implica tener alguna experiencia previa manejando un soldador. Debemos trabajar sobre el circuito de la misma computadora, y dado el poco espacio que hay en él, hay que tener mucho cuidado de no hacer un corto circuito o cortar alguna línea del mismo mientras se cambian algunos componentes. Procuraremos un soldador de aproximadamente 35 watts, un alicate, un par de pinzas y muy buen pulso.
El trabajo de convertir la máquina no nos llevó más de media hora.
Una vez terminado el mismo, procedimos a probar los resultados y seguir las instrucciones de calibración del fabricante.
Pudimos ver de inmediato los distintos colores aunque estaban algo saturados y el blanco no era realmente blanco.
Siguiendo las instrucciones de calibración que acompañan al kit, ajustamos dos presets de la máquina hasta obtener los colores adecuados.
Como conclusión, podemos decir que ésta es una buena opción para todos aquellos que tengan alguna experiencia soldando y ganas de ver los colores que siempre supieron que estaban pero nunca se dieron a conocer■

ANALIZADOR DE CODIGOS EN LENGUAJE DE MAQUINA Z-80

Juan Pablo Luccioni

II parte

Dada la excelente calidad y la aceptación que tiene el programa de nuestro "PROGRAMADOR DEL AÑO", aquí van las explicaciones detalladas de este utilísimo desarrollo.
Muchos programadores avanzados y no tanto han pedido las explicaciones completas del programa de Juan Pablo Luccioni. Rápidamente se compenetrarán en las explicaciones que el mismo Juan Pablo escribió para nuestra revista.
Pero, indudablemente, dada la claridad de éstas, será de mucha utilidad para los que están dando los primeros pasos en el impenetrable Assembler.

EXPLICACION DEL FUNCIONAMIENTO ESTRUCTURAL

El programa está formado por dos tablas, un área de trabajo y el código de operación propiamente dicho. El área de trabajo está ubicada en el interior del programa y corresponde al largo conjunto de ceros que se observa en el listado hexadecimal que se provee.

Una de las tablas, de 256 bytes, altamente optimizada para albergar toda la información del procesador, se utiliza para conocer, dado el opcode de una instrucción (que representa la clave de acceso a la tabla), su longitud y algunas propiedades que están relacionadas con el conjunto de instrucciones del Z-80. Accediendo a la tabla con el opcode se obtiene un valor de 0 a 225 que se forma de la siguiente manera:

— **bit 0:** siempre a 0.
— **bits 1, 2, 3, 4:** representan los siguientes valores

- 2 instrucción de un byte
- 4 instrucción de dos bytes
- 6 instrucción de tres bytes
- 8 algún tipo de CALL
-10 algún tipo de JP
-12 algún tipo de JR
-14 algún tipo de RST
-16 instrucción DJNZ
-18 algún tipo de RET
-22 JP (HL), JP (IX), JP (IY)

— **bit 5:** a 1 implica que se trata de cambio de modo de interrupción si el opcode sigue a 'ED'
— **bit 6:** a 1 implica que si el opcode anterior es 'ED', la instrucción tiene una longitud de 4 bytes y no los indicados por los bits 1, 2, 3, 4
a 0 implica que si el opcode anterior es 'ED', la instrucción tiene 2 bytes.
— **bit 7:** a 1 implica que si el opcode anterior es "DD' o 'FD', la instrucción tiene 2 bytes **más** que los que indican los primeros bits.
a 0 implica que si el opcode anterior es 'DD' o 'FD', la instrucción tiene 1 byte **más** que los especificados por los primeros bits.

La segunda tabla es de 25 bytes, y contiene las direcciones de las subrutinas, dentro del programa, que corresponden a las distintas condiciones expresadas por los bits 1, 2, 3, 4 de la tabla 1. De esta manera, se puede acceder a esta tabla con el valor que representan los bits 0, 1, 2, 3, 4 obtenidos de tabla 1 (un valor siempre par porque cada dirección en la tabla 2 ocupa 2 bytes).
Ambas tablas se encuentran ubicadas a partir de un byte 0 de página, de tal manera de optimizar al máximo el acceso a las mismas. La primera tabla comienza en ORIGEN + 256 y la segunda comienza en ORIGEN.

Esta es la razón por la cual todas las versiones del intérprete comienzan en una dirección divisible en 256.

Antes de seguir, vamos a exponer la clasificación de las instrucciones del Z-80 a los fines de este programa:

1.— Instrucciones de TRANSFERENCIA: representan los códigos 8 al 22 en la tabla 1, deben ser tratadas en forma particular por subrutinas específicas, cuyos nombres son EJECJUMP, EJECALL, EJECRET, EJECRESTART, EJECRELJUMP y EJECDJNZ y cuyas direcciones figuran en ta-

ACLARACION

La primera parte de esta nota fue publicada en nuestro número 18, del mes de SEPTIEMBRE 1986. Si desafortunadamente no tenemos este ejemplar en nuestro poder, basta con solicitarlo por correo con el giro postal correspondiente, o bien, pasar a retirarlo por nuestra casa, Paraná 720, 5° piso, Capital Federal.

bla 2 accediendo con el valor de tabla 1.

2.— Instrucciones de EJECUCION LINEAL: En un proceso normal son ejecutadas una tras otra, no producen nunca una bifurcación del programa como su consecuencia. Pueden tener 1, 2, 3 o 4 bytes de largo. Para su simulación se utiliza un método que es la parte **especial** de este programa. Una vez determinada su longitud, la instrucción se copia en un **BUFFER** de **EJECUCION**; haciendo pasar luego el procesador Z-80 (real) a través de ellas, no sin antes descubrir todos los registros del proceso que se simula y teniendo la precaución de resguardarlos inmediatamente después y sin introducir modificaciones. Las instrucciones idóneas para estos fines forman el grupo PUSH/POP. En el stack de la máquina (aquel mismo que utiliza el intérprete BASIC), se guardan los registros del proceso (programa que se simula), por lo que se habilita además un seudo stack, que servirá como stack del proceso y cuya dirección es completamente variable y se altera con el mismo programa o con instrucciones CALL, RET. El buffer de ejecución siempre tiene 4 bytes, siendo los últimos bytes 0 (NOP) si la instrucción lineal tiene menos de 4 bytes.

La Figura 1 aclara el manejo del buffer.

Las instrucciones de bifurcación son tratadas en forma particular por subrutinas que han sido optimizadas a través de la idea de automodificación, basada en la conversión de instrucciones JR condic; CALL condic; y RET condic a su correspondiente JP condic, es decir a un Opcode de Jump que tenga la misma condición de decisión. Se descubre el registro AF del proceso justo antes de pasar por la instrucción JP condic y basado en esta decisión, se simula la actuación de cada una de estas instrucciones por separado. Esto evita preveer cada tipo de condición de bifurcación por separado, lo que hubiera sido muy largo de programar.

Existe un sector del programa, que comienza con la etiqueta DRIVER, que representa uno de los extremos en el lazo de ejecución, instrucción por instrucción, y obtiene el código de operación accediendo a ambas tablas para luego saltar a las subrutinas correspondientes.

Para una instrucción de ejecución lineal se pueden presentar dos posibilidades:

a) Instrucciones NORMALES: no incluyen manejo de registros IY, IX ni códigos de extensión. Son las que más directamente se procesan.

b) Instrucciones EXTENDIDAS: cuya longitud es calculada a través del siguiente byte de la instrucción, con la información de tabla 1.

Existe una subrutina que emite un sonido cuando se ejecuta una instrucción de cambio de modo de interrupción; aunque en realidad ésta ha sido incluída para que emita un sonido cada vez que aparece un valor especial de opcode, que viene dado por el bit 5 de tabla 1; característica muy útil para futuras ampliaciones o aplicaciones específicas del programa.

Existen fundamentalmente tres sectores del programa por donde comienza o termina el proceso:

— **START**, que realiza toda las inicializaciones previas y cuya dirección es la que llamanos INICIO.

— **REENT**, cuya dirección corresponde es REINICIO, es el lugar a donde se accede después de cumplir un servicio del usuario en BASIC, y tiene un proceso de reinicialización muy diferente a START.

— **ERROR**, es el nombre de la estructura de salida, una vez que se ha detectado una secuencia de teclas BREAK.

VARIABLES DEL ENSAMBLADOR

Debe notarse que todas las variedades del archivo fuente son relativas a la dirección ORIGEN, que es el primer byte del programa, lo cual facilita el proceso de reubicación.

— **TABLAJUMP** es el número de página de ORIGEN, es decir el byte alto de la dirección de la Tabla 2.

— **TABLA** es el byte alto de la dirección de Tabla 1, que está desplazada 256 bytes con respecto a Tabla 2.

— **STACKPROC** es una dirección que especifica el lugar donde se pone originalmente el stack del proceso que se simula. Este valor se especifica como el tope del

seudo stack cuando comienza el programa por primera vez. Esta dirección es interna al programa (ORIGEN +220).

— **CODE** es una dirección donde se almacena el valor obtenido de Tabla 1.

— **PC** son dos direcciones donde se almacena el program counter del proceso que se simula.

— **SPPROC** es una dirección donde se encuentra el valor tope del seudo stack, originalmente toma el valor stackproc, pero su contenido puede ser modificado con instrucciones de manejo de stack del programa que se interpreta.

— **SPFILL** es una dirección que especifica el valor tope del stack de la máquina que se utiliza para resguardarlo cuando se cambia del stack de la máquina del seudo stack del proceso.

— **INS** es una dirección donde se coloca el opcode que se está ejecutando. Es útil para consultar desde BASIC (ORIGEN +40).

— **DIEI** es una dirección donde se resguarda el contenido de la báscula de interrupciones del Z-80 (IFF1) cuando el proceso retorna al BASIC.

DESCRIPCION ESTRUCTURA POR ESTRUCTURA

Se sigue un orden lógico y no el que aparece en el archivo fuente, pero los nombres son los mismos:

START: Corresponde a la dirección de INICIO.

La primera operación realizada es poner el valor correcto en SPPROC y luego hacer una copia del stack de la máquina en la parte superior de la dirección especificada por SPPROC, lo que tiene por objeto permitir la ejecución de una instrucción RET de retorno al BASIC si ésta existiera en el programa a interpretar.

Luego se resguarda el contenido de HL' cuyo valor es esencial al tiempo de retornar al BASIC con RET, y se llena el stack con valores 23610, que corresponden al estado inicial de los registros del proceso. Esto es una simplificación, ya que por lo menos el registro ly debe contener ese valor. Como HL' había sido resguardado de todas maneras), para permitir retornos al BASIC mediante la situación de una instrucción RET. Se pone en la variable PC el valor de SEEDE1 proceso sigue por la estructura DRIVER*

DRIVER: Su función principal es obtener el opcode de la dirección especificada por PC, acceder a Tabla 1, enmascarar el valor obtenido y acceder a Tabla 2. Existe una gran simplicidad en este procedimiento gracias a la optimización del sistema de tablas. Obtenido en HL la dirección de la subrutina que hará el servicio de este opcode, de salta con la instrucción JP (HL).

Este es uno de los puntos que dificultan el entendimiento de este programa si no se conoce el contenido de las tablas, ya que no existe una conexión directa entre la estructura DRIVER y las estructuras de servicio.

NORMAL: Representa la subrutina de servicio de las instrucciones de ejecución lineal, se determina la longitud de la instrucción y se la copia en el buffer de ejecución, adelantando el PC de manera conveniente. Después se salta a la estructura EXEC. Existen dos entradas externas a esta estructura que son NORMAL 1 y NORMAL 2, y son usadas por las subrutinas de instrucciones EXTENDIDAS (IX, IY, etc.), de tal manera que todo volcamiento sobre el buffer se realiza por la estructura NORMAL. Se denomina LINEA a la dirección del primer byte del buffer, dirección que calcula el ensamblador.

EXTENDED: En esta estructura se procesan aquellas instrucciones que comienzan con ED y cuyas longitudes deben calcularse en base a los bits superiores en combinación con los inferiores de Tabla 1. Esta estructura siempre desemboca en NORMAL 2, de tal manera que utiliza la parte de transferencia al buffer de NORMAL, pero los valores de longitud son calculados por separado. Existe una bifurcación por subrutina a WARNING, cuando se detecta el bit 5 del valor de Tabla 1, en particular esto ocurre en instrucciones de cambio de modo o RETI, RETN; pero esta situación puede ser alterada en futuras versiones cambiando la Tabla 1.

WARNING: emite un sonido con cambio de color del borde. Sólo reacciona con un tipo particular de instrucciones que tienen el bit 5 en Tabla 1 a 1. De esta manera, un programador interesado puede hacer que se ejecute con las instrucciones que desee.

INDEX: En esta estructura se procesan instrucciones que utilizan los registros índices (IX, IY), se produce la detección de los JUMP (ix), (iy), se calcula las longitudes correspondientes y se salta a NORMAL 1, de tal manera que la transferencia al buffer de la instrucción indexada también es realizada por NORMAL. *SEED es la variable del BASIC donde se pasa el argumento de Pc.

EXEC: Es la subrutina de ejecución propiamente dicha. Se accede a ésta después de haber volcado en el buffer (que forma parte de EXEC) la instrucción a ejecutar. En primer lugar se descubren todos los pseudoregistros del proceso mediante las instrucciones POP correspondientes (recordar que en el stack de la máquina se guardan los pseudoregistros), luego se guarda en SPFILL la dirección tope de la pila de máquina y se asigna al registro SP el contenido de SPPROC (tope de la pseudopila). La próxima instrucción que se ejecuta ya pertenece al buffer (etiqueta LINEA) y está seguida de NOPs hasta completar los cuatro bytes, según la longitud que tenga (recordar que los NOPs no afectan ningún registro). Una vez ejecutada esta instrucción que había sido traída del programa que se simula, se realiza todo el proceso inverso: se resguarda el valor tope de pseudostack, se descubre el stack de la máquina y se realiza el PUSH DE los registros que corresponden al proceso. Con dos instrucciones se 'borra el buffer' (se le pone ceros). Se continúa el proceso por la estructura FINAL.

FINAL: Se realiza la lectura del teclado y se salta a DRIVER (si no se debe retornar al BASIC) o ERROR si se ha apretado la secuencia de teclas BREAK.

ERROR: Es la única extructura de salida al BASIC. Antes de retornar, se guardan todos los registros del proceso, que están en el stack de la máquina, en direcciones de memoria intermedia al programa (aquellas que se consultan desde BASIC), se restituye el valor de HL' y se averigua con la instrucción LD A,I el contenido de IFF1 (báscula de interrupciones del procesador), modificando adecuadamente la variable DIEI. Esto se debe hacer para resguardar para el reinicio la condición de interrupciones, ya que se debe ejecutar EI antes de retornar al basic. Antes de hacer esto último, se carga en el registro BC el contenido de las direcciones PC, para enviar información directa al BASIC sobre la dirección en ejecución.

REENT: Es la estructura cuya dirección corresponde a REINICIO, en ella se vuelcan en el stack de la máquina el valor de los registros del proceso que se habían guardado en memoria en ERROR y eventualmente fueron modificadas por el usuario en BASIC. Se toma las precauciones expuestas con el registro HL. Finalmente se coloca el modo de interrupción que había antes de retornar al BASIC mediante el dato de la variable (DIEI). Luego se salta a la estructura DRIVER.

ESTRUCTURAS DE MANEJO DE INSTRUCCIONES DE BIFURCACION

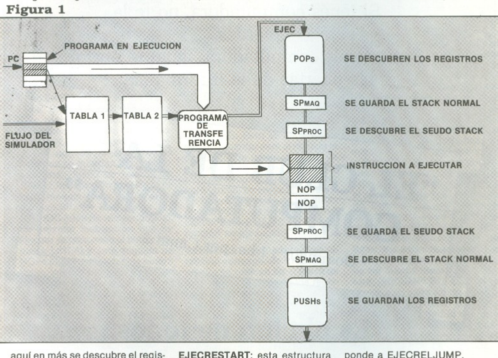
A todas estas líneas se accede con el JP (HL) que se realiza en DRIVER de acuerdo con los datos de Tabla 2.

EJECJUMP: se toma el código de JP condición y se lo inserta en el programa (automodificándolo) en la posición LOADJUMP, que está unos pocos bytes más adelante. Se descubre el registro AF del proceso (con un simple POP, PUSH, recuérdese que este registro siempre es la cola del stack pointer de la máquina) y la próxima instrucción en ejecutarse es el Jump que corresponde al del programa, es decir, responde a la misma condición de los flags (F), y éstos han sido descubiertos justo antes; pero la dirección del salto no es aquélla que figura en el programa que se está simulando, sino una fija que corresponde a la condición de cumplimiento del salto (SECUMPLE1). A partir de aquí existen dos subestructuras: una a continuación del jump y que corresponde a la condición de no cumplimiento del salto y otra (SECUMPLE1) que corresponde a un salto válido. En la primera, se incrementa el valor del PC 3 veces (longitud de la instrucción JP) y se bifurca a FINAL (la misma estructura a la que bifurcada EXEC). La segunda, es decir SECUMPLE1, toma de los dos bytes siguientes del programa que se simula, la dirección a la que hay que saltar y la copia en PC, luego salta a FINAL.

EJECCALL: Estructura que se encarga del manejo de los CALL condición, que se convierten, restándole 2 al opcode de un opcode de JP con la misma condición, y se carga este valor en LOADCALL (automodificación). Antes se había filtrado el CALL NN que no sigue la regla de restar 2. De principio que en ejeccall. Sumando 2 al opcode de RET condición se transforma en el opcode de un JP con la misma condición, que se vuelca al programa en LOADRET. Por lo demás, se sigue las mismas operaciones que haría con el stack el procesador cuando ejecuta un RET.

EJECDJNZ: Es una estructura que simula la actuación del procesador en la ejecución de un DJNZ disp. Lo primero a hacer es consultar el registro B, que está en la pila de la máquina y decrementarlo. En caso de requerirse la bifurcación, utiliza la estructura SECUMPLE4, que corresponde a EJECRELJUMP.

Figura 1



aquí en más se descubre el registro AF del proceso y la próxima instrucción a ejecutarse es el Jump que tiene la misma condición del Call del programa pero su dirección de salto es SECUMPLE2. Si no se cumple el salto, la operatoria es la misma que para el caso de jump; si se cumple el salto, se coloca en las direcciones correspondientes que señala SPPROC la dirección especificada por el PC en el programa que se está ejecutando, se pone el valor obtenido del Call en la variable PC y se actualiza convenientemente SPPROC. Esta operación es la misma que realiza el procesador cuando ejecuta un Call.

EJECRET: Se sigue el mismo

EJECRESTART: esta estructura cumple el servicio de los RST NN, que no son más que CALL NN con direcciones implícitas, que se pueden calcular restando al opcode el valor 199. Por lo demás, se opera como lo haría el procesador.

EJECRELJUMP: esta subrutina trata los JR condición, automodificándose directamente con el opcode de la instrucción. En caso de darse la condición, salta a SECUMPLE4, donde el nuevo valor de PC se calcula teniendo en cuenta que el valor que sigue al opcode de JR está en complemento de 2 y respresenta el desplazamiento sobre el actual PC. La estructura COMPLEMENT se encarga de los saltos negativos.

JUMPHL: Procesa la instrucción JP (HL), averiguando el contenido de HL del proceso y poniéndolo como PC.

JUMPIXIY: esta estructura es llamada desde INDEX, y procesa, de manera semejante a JUMPHL, las instrucciones JP (IX) y JP (IY). El programa que se encuentra al final del source file para Zeus Assembler tiene como finalidad crear la Tabla 2 en base a las direcciones de las estructuras que han sido expuestas: NORMAL, EJECXXXX, INDEX, EXTENDED, etcétera. Luego transmite desde 32500 hasta origen + 256 la Tabla 1. Este programa no tiene nada que ver con el intérprete en sí mismo■



EDUCATIVO

• Primeros pasos y diversos proyectos (Congreso en el Sheraton)
• Programas desarrollados por los docentes (Escuela del Sol)

PROGRAMAS INEDITOS

• Aprobando los exámenes
• Reloj mundial
• Calorías
• Repaso

UNIVERSIDADES ARGENTINAS Y EXTRANJERAS INTER-CONECTADAS

Un vasto programa alienta la investigación y la capacitación en el nivel primario, secundario y terciario. Ahora apunta a ligar a nuestro país con una red académica mundial.

El Licenciado Adolfo D'Onofrio, gerente de programas científicos, y Carlos Sanjurjo, gerente de Relaciones Externas de IBM Argentina, fueron los encargados de informarnos cómo la empresa está apuntalando la inserción de la informática en el plano educativo nacional.

Este tipo de decisión se cristalizó en 1980 con el primer convenio firmado entre IBM y la Universidad de Buenos Aires. El acuerdo posibilitó la creación del Centro de Tecnología y Ciencia de Sistemas —CTCS—. La empresa colocó una computadora importante, la 370 Mod.158 de 4 Megabytes, que todavía está instalada. Y ahora se está cambiando por una 3031 de 8 Megabytes.

"Ese Centro tuvo una gran importancia dentro de la Universidad porque perseguía dos objetivos: uno, desarrollar grupos de investigación asistidos por computadoras dentro de la casa de estudios, y el otro, capacitar funcionarios de la Administración Pública y personal de la Universidad en técnicas avanzadas de computación" especificó Sanjurjo.

En relación a la Administración



El director nacional de Asuntos Universitarios, doctor Hugo Storani y el rector de la Universidad Nacional de Río Cuarto, ingeniero Roberto Seiler proceden a dejar inaugurado el Sistema IBM/370 modelo 3031. Más atrás, aparecen Adolfo D'Onofrio (izquierda) y a su lado Carlos Sanjurjo.

Pública, la capacitación abarca toda la escala de funcionarios, desde la Municipalidad más pequeña hasta personal de la Presidencia de la Nación. La educación se da sobre técnicas avanzadas de procesamiento de datos.
Pero el convenio también apuntó a una verdadera difusión de la informática en profesiones tradicionalmente no informatizadas. Adolfo D'Onofrio explicó que "hay cursos que se realizan específicamente para abogados, médicos, odontólogos, docentes, que se efectúan desde hace varios años, en los que no se enseña informática educativa sino técnicas para gente que está interesada en buscar la aplicación dentro de su profesión. Se hizo una difusión muy grande y han pasado 2600 alumnos en los últimos cinco años".
En relación a los grupos de investigación hay 175 trabajos en marcha. Cabe destacar que el Centro dio cabida a grupos de distintas facultades para trabajar en las terminales.

Aplicaciones Profesionales

"Hay una cosa muy curiosa en el trabajo de investigación. Cuando se habla de ello, uno piensa que es la gente de Ciencias Exactas o Ingeniería la que llevó más trabajos. Sin embargo, la mayor parte de esas tareas no son de esas facultades. Por ejemplo, una parte muy importante viene de Agronomía" resaltó Sanjurjo "Realmente captaron la importancia de la computación dentro de una profesión a la cual no se había informatizado mucho y oficialmente no había intención o esfuerzo por informatizarla. Ellos mismos se fueron interiorizando de las necesidades que podían ser cubiertas por la computación. Un porcentaje muy grande de los trabajos, en estos momentos viene de Agronomía y Veterinaria y tal es así, que uno de los Centros Remotos que tiene el CTCS está en Agronomía".


Consejo Nacional de Educación (CONET)

Cuando IBM comprendió que el volúmen de tareas produce un gran flujo de investigadores, el CTCS armó Centros Remotos y en la actualidad Ciencias Económicas, Agronomía y uno pequeño en el Hospital Escuela —en el que se hace entrenamiento de personal médico por medio de computadoras— funcionan con éxito.

Laboratorios

Otro proyecto importante se realizó entre 1984 y 1986. Ante la euforia general de la computación en las escuelas, pero sin una idea bastante clara de cómo implementarla, IBM no quiso llegar a los establecimientos educativos sin antes hacer una verdadera investigación pedagógica con grupos interdisciplinarios. De esa manera se podría ver de qué manera realizar una mejor inserción de la informática en la educación. Así, se crearon laboratorios en sociedad con la Municipalidad de Buenos Aires, con la Universidad Católica Argentina, con la Universidad Católica Santo Tomás de Aquino de Tucu-

mán, con una fundación privada —FECIC—, y se instalaron dos laboratorios en la facultad de Ciencias Exactas de la U.B.A., uno en el centro de Computación y otro en el departamento de Química Orgánica.
También hay un laboratorio de informática educativa en La Plata en la Fac. de Ciencias Naturales. Lo realizado en esta casa de altos estudios es sumamente novedoso por tratarse de una disciplina en la que no había entrado la computadora, por lo menos en Argentina. Allí mantienen un con-



Universidad Católica Argentina

tacto muy cercano con investigadores de Estados Unidos, que están trabajando en temas similares.
El grupo argentino es muy entusiasta y están desarrollando aplicaciones directas en las ciencias naturales, arqueológicas, antropológicas, clasificación de seres vivientes con técnicas de taxonomía numérica, etc. Sanjurjo aclara que los grupos de experimentación en la enseñanza los tienen en los tres niveles. En el primario: con la Municipalidad de Buenos Aires. A nivel secundario, el convenio más antiguo que tienen es con el CONET, con resultados muy buenos en cuanto a la creación de soft educativo y en el universitario, con varias universidades nacionales y privadas. Los programas desarrollados con el CONET se van a publicar para que se pueda apreciar algo que realmente es una guía en cuanto a la aplicación de la informática en la enseñanza secundaria.
Algunos de estos programas pueden también ser usados en el campo universitario. Uno de los objetivos del proyecto educativo es que cada laboratorio tenga una orientación distinta. Algunos están trabajando sobre materias físico-matemáticas, otros sobre humanísticas y otros en ciencias naturales, cubriéndose así las diferentes áreas.

CENTROS DE COMPUTOS



El entrenamiento de funcionarios públicos y el funcionamiento de laboratorios de informática educativa ya son un hecho. Sin embargo, la idea de implementar proyectos conjuntos con el Estado no se agota. Nos explican que un tercer aporte consistió en que "nosotros tomamos nueve unidades centrales de proceso que estaban ubicadas en alquiler en empresas del Estado y le propusimos al Ministerio de Educación, en el momento en que el gobierno las reemplazase agregarles el periférico necesario —cintas, discos, impresoras y otros— para formar nueve centros de cómputos con 18 terminales cada uno, que se donaron con la única condición de que fueran instaladas en universidades nacionales. Además se donó el mantenimiento por un cierto lapso y el software adecuado. Eso se presentó al Ministro de Educación simplemente porque IBM no quiso seleccionar por sí mismo las universidades que las recibirían. De las nueve, ya se instalaron seis, la sexta en Río Cuarto y hay dos muy próximas a instalarse en la UTN (Bs. As.) y la Universidad de La Plata; aquí, una 3032, una máquina más grande. Faltaría la última, pero aún no se firmó el acuerdo correspondiente". Este ambicioso plan se completa con las máquinas instaladas en el CTCS, en total diez grandes sistemas constituyendo el reequipamiento informático académico más importante que ha hecho el país hasta este momento.
Las perspectivas futuras son alentadoras y en IBM ya las están evaluando. Si diez universidades cuentan con centros de cómputos importantes, el próximo paso es constituir la red universitaria nacional.
Se tratará de desarrollar una red académica exclusivamente interuniversitaria. Esas diez universidades —las nueve más el Centro de Tecnología— quedarían ligadas en una red nacional. Eso permitirá intercambiar información entre ellas, compartir bancos de datos y poder ser conectadas a las redes académicas internacionales, lo que constituirá el próximo paso de envergadura.

RED UNIVERSITARIA MUNDIAL

Existe una red mundial de universidades que cubre desde Israel hasta Japón, pasando por toda Europa, Norteamérica, Canadá y México.
Fue impulsada inicialmente en Estados Unidos por IBM y desarrollada en conjunto con institutos de investigación y casas de altos estudios. En la actualidad cubre más de 430 universidades en el mundo (más de 200 en Estados Unidos).
Ahora el objetivo es ligar un nodo argentino, que seguramente va a ser el CTCS, a un nodo de entrada a la red americana, para de ahí irradiarse a todas las demás universidades. Cada uno,

por razones de economía de comunicación, debe acoplarse al nodo geográfico más cercano que tiene, o de más frecuente comunicación. Entre las universidades argentinas y estadounidenses (fundamentalmente de California) hay un gran intercambio. Sobre todo por argentinos que están trabajando allá. Así la universidad de California sería el corresponsal adecuado.

La red de ningún modo es limitativa. IBM conectó sus máquinas a la red y luego cada compañía ha hecho lo mismo para entrar en ella. En estos momentos hay una enorme cantidad de diferentes marcas usufructuando su uso. Son casi 2.000 máquinas en el mundo.

Lo importante es que cada gran computadora tenga en torno suyo y distribuidas dentro de la universidad una cantidad muy grande de pantallas. Esta red permite que desde cualquier pantalla se direccione otra cualquiera del mundo y se intercambie un mensaje directamente de investigador a investigador.

Fundación Banco Boston

La implementación de este sistema posibilita, por ejemplo, el trabajo tipo forum, es decir que si una persona está interesada en un tema puede escribir en una especie de pizarrón de noticias "soy fulano de tal, trabajo en tal universidad, estoy interesado en tal tipo de tema y me gustaría ponerme en contacto con todos los que trabajan en temas similares". Luego se puede revisar periódicamente esa especie de casilla de correo y encontrar diferentes mensajes. Es un sistema que abre las puertas a la supercomunicación académica.

El paso, de darse, sería trascendental. Se ligaría el mundo académico argentino al mundial.

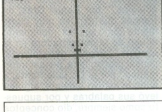


GRAPH

COMP.: CZ 1000/1500; TK 83/85
CONF.: 2 K
CLAS.: EDU

Permite graficar varios puntos en la pantalla. Los ejes coordenados se ubican automáticamente.
Hay que ingresar los valores de X y de Y, tantas veces como puntos se desee graficar.
Cuando se haya terminado de entrar los datos, se tipea −9999; si los valores son muy grandes en un eje, cambia automáticamente la escala.





```
  2 DIM X(17)
  3 DIM Y(17)
  4 PRINT "K-64 COMPUTACION PAR
A TODOS"
 10 LET J=1
 20 PRINT AT 2,5;"  X        Y"
 30 PRINT AT 20,0;"ESCRIBA LA C
OOR. X"
 40 INPUT X(J)
 44 IF X(J)=-9999 THEN GOTO 200
 45 PRINT AT 2+J,5;X(J)
 50 PRINT AT 20,17;"Y"
 60 INPUT Y(J)
 70 PRINT AT 2+J,12;"█  ";Y(J)
 80 LET J=J+1
 90 GOTO 30
200 LET A=30
205 CLS
210 LET B=20
220 LET C=-30
230 LET D=-15
235 FAST
240 FOR F=1 TO J-1
250 IF X(F)>A THEN LET A=X(F)
260 IF X(F)<C THEN LET C=X(F)
270 IF Y(F)>B THEN LET B=Y(F)
280 IF Y(F)<D THEN LET D=Y(F)
290 NEXT F
310 LET XS=(A+ABS C)/60
320 LET YS=(B+ABS D)/35
321 FOR F=1 TO J-1
322 LET V=C
324 LET T=D
325 IF X(F)=A THEN LET X1=60
326 IF Y(F)=B THEN LET Y1=41
327 LET M=0
328 LET N=0
330 IF T+YS>Y(F) THEN LET Y1=AB
S (D-T)/YS
340 IF V+XS>X(F) THEN LET X1=AB
S (C-(V-XS))/XS
350 IF T+YS>Y(F) THEN LET M=1
360 IF V+XS>X(F) THEN LET N=1
370 IF M=1 THEN GOTO 390
380 LET T=T+XS
385 GOTO 330
390 IF N=1 THEN GOTO 420
400 LET V=V+XS
410 GOTO 340
420 PLOT X1,Y1
430 NEXT F
440 SLOW
450 LET T=INT (D/YS)
460 LET P=ABS T
465 IF XS>1 AND YS=1 THEN LET P
=P-1
470 LET T1=INT (A/XS)
480 LET P1=59-T1
490 IF XS>1 THEN LET P1=P1-1
500 FOR J=1 TO 60
510 PLOT J,P
520 NEXT J
530 FOR J=5 TO 40
540 PLOT P1,J
545 NEXT J
550 PRINT
550 PRINT "X=";XS
570 PRINT "Y=";YS
580 PAUSE 4E4
590 CLEAR
591 CLS
600 GOTO 0
1000 SAVE "GRAPH"
1001 RUN
```

PRIMEROS PASOS Y DIVERSOS PROYECTOS

Dos mil docentes participaron en el congreso realizado en Buenos Aires. Expusieron disertantes de primer nivel, mientras que profesores y alumnos comentaron sus experiencias.



Profesora Xargay de González junto con sus alumnos.

Con el lema "La conjunción entre educación y progreso técnico se proyecta al servicio del hombre", se realizó el Primer Congreso Argentino de Informática Educativa, con el auspicio de instituciones oficiales y privadas. En el Sheraton Hotel se dieron cita unos dos mil docentes que llegaron de diferentes lugares de nuestro país. Algunos, ávidos por escuchar y ver experiencias nuevas en este tema y otros, ya conocedores y profesionales en la materia, disertaron contando los caminos seguidos y nuevas propuestas.

El Congreso abarcó cinco días hábiles, con sus mañanas y tardes en forma completa. Hubo disertaciones académicas en sesiones plenarias, conferencias sectoriales. Se realizaron talleres de demostración práctica, vimos también stands pertenecientes a empresas dedicadas a la Informática que, ubicados en el primer piso del hotel, nos deslumbraron por la calidad de sus productos: hardware y software, y por último, lo que realmente encandiló los ojos de los visitantes fueron los trabajos realizados por los alumnos de varias escuelas expositoras.

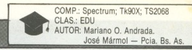
Con respecto a las sesiones plenarias, las hubo de primer nivel, no sólo por los temas importantes por las cuales transitaron, sino también por los disertantes de primera línea que expusieron. Les damos algunos títulos: "La Informática Educativa: ¿un nuevo instrumento o un cambio educativo de fondo"? por el Dr. Oscar Gomez Povina. "¿Cómo implementar la Informática en los colegios?" por Daniel Gras, un francés que expuso muy claramente cómo su país implementó el uso de la computadora en la escuela. "Pensar determinista, pensar probabilista, pensar informático" por el Dr. Luis Santaló; aunque estuvo ausente, alguien leyó sus palabras y por supuesto como siempre, dio conceptos claros y contenidos muy valiosos. También estuvo presente la profesora Nelly Vázquez de Tapia con el tema "El computador en el aula, un interlocutor inteligente". Y otro Plenario que atrajo mucho público por la importancia del tema tratado fue el del Ing. Daniel Tkach con el tema "Inteligencia artificial y educación".

Las conferencias sectoriales tuvieron diferente forma, ya que se hicieron como presentación de trabajos, como panel o como exposición, es decir que hubo variedad en los temas tratados y sus enfoques. Aquí un grupo de escuelas contó la experiencia en infomática educativa, cada una con su realidad pero con la honestidad y valentía de un grupo de docentes que dijeron cómo comenzaron esta experiencia de la inserción de la Computación en la escuela, cómo dieron los primeros pasos, qué hacen en la actualidad y esbozaron proyectos para el futuro.

Las escuelas que se hicieron presentes con sus rectores, grupos de docentes y alumnos fueron:

Belgrano Day's School, Goethe, La Salle, Santa Brígida, Lincoln (Ituzaingó), Escuela Argentina Modelo, Nuestra Señora de las Nieves, Northland, Cardenal Newman, Sagrada Familia (Tandil) y Escuela Normal Nacional Olegario V. Andrade (Gualeguaychú).
En estas conferencias se desarrollaron variados temas, como "La formación de docentes en informática educativa" (el salón se colmó de público). Muy interesante fue escuchar los proyectos a nivel nacional que se están realizando para la implementación de la Informática en la educación argentina; éstos son el Proyecto Constelación de UTN y el Proyecto Andes de Dinem.
En los talleres los expositores desarrollaron diversos temas de la currícula estudiantil de todos los niveles, con la utilización de microcomputadoras a las que tenían acceso los participantes. Algunos de los temas que se trataron fueron "Trigonometría y computadora", "Música y computadoras", "Software educativo en la escuela".
También hubo una muestra donde se observó hardware de muy buena calidad y software (en general extranjero). Amstrad exhibió hardware y un software de aplicación bastante amplio. El stand de Talent fue realmente muy visitado, su computadora TALENT-MSX fue observada con mucho interés por directivos y docentes de escuelas que desean insertar a la computadora como un nuevo recurso didáctico en sus establecimientos; además es digno de mencionar que TALENT está auspiciando la producción de software educativo y anunció que muy pronto estará disponible, por supuesto compatible a dicha norma.
También Czerweny exhibió sus computadoras CZ y Spectrum, y pudieron verse accesorios (como los datassette Mitsao).
Editorial Proedi, editora de las revistas "K64. Computación para todos", "Load MSX" y revista para usuarios Drean Commodore, estuvo presente con un stand.
Pero realmente "el dulce" del Congreso fue la muestra de trabajos que ofrecieron los alumnos de las diferentes escuelas participantes. Los grupos de alumnos que, frente a las computadoras parecían verdaderos directores de orquesta, comentaron cómo realizaban sus trabajos y la verdad es que la creatividad y la calidad se pusieron de manifiesto en todo momento. Merecieron por esto el cálido aplauso de todos los visitantes. Por ejemplo, estudiantes de la Escuela Argentina Modelo pertenecientes a los cursos superiores del nivel secundario contaron cómo realizan programas educativos para ser utilizados en la computadora por alumnos del mismo establecimiento, pero de jardín de infantes y grados inferiores del nivel primario. ■

Nuria Duran Xargay de González



APROBANDO EXAMENES

Este programa puede almacenar, modificar, consultar y listar datos en forma de preguntas y respuestas.

El mismo permite almacenar 50 preguntas con sus respectivas respuestas en todas las materias, obteniendo resultados altamente positivos que quiero compartir. Las preguntas no pueden tener más de 96 caracteres y las respuestas no pueden tener más de 288 caracteres.

Como éstas últimas suelen ser más extensas que las primeras, permiten acumular respuestas más completas y por consiguiente más cantidad de datos, ya que el ordenador está limitado por sus 48 K.

A pesar de dicha limitación el programa cuenta con importantes ventajas.

1) Indica la cantidad de datos usados y la cantidad de datos libres.

2) Alta velocidad de selección de datos.

3) Protección para que no se puedan ingresar más datos que los dimensionados en el programa.

4) Las opciones 1, 2, 3, 4, y 5 poseen una tecla de escape "=", que al presionarla, el programa retorna al menú principal.

5) Oprimiendo DELETE se eliminan los caracteres equivocados.

MENU PRINCIPAL

1) Ingresar datos
2) Modificar datos
3) Consultar datos
4) Ejercitación
5) Listado completo
6) Grabar datos en cinta
7) Fin del programa

INSTRUCCIONES PARA OPERAR EL PROGRAMA

Opción: 1) "Ingresar datos"
Al presionar la opción 1 seguida de ENTER, aparecerá en la pantalla el título y un cursor pidiendo que se ingrese la pregunta. Una vez ingresada la pregunta se debe presionar ENTER.Hecho esto, la pregunta quedará almacenada en el computador. Luego aparecerá nuevamente el cursor pidiendo que se ingrese la respuesta, en este caso se procede de la misma forma que en el paso anterior. Al ingresar la respuesta el programa vuelve al menú principal.

Opción: 2) "Modificar datos"
Al seleccionar la opción 2 el programa le pedirá que decida si desea modificar la pregunta o la respuesta. Para hacer esto sólo es necesario presionar el número de opción (1 o 2). Hecho esto el computador le indicará qué es lo que debe hacer; luego la computadora buscará el dato deseado y cuando lo encuentre le dejará el cursor preparado para que se efectúe la corrección.

Opción: 3) "Consultar datos"
Con sólo ingresar la pregunta, seguida de ENTER, la computadora buscará la respuesta.
Si la pregunta ingresada difiere, aunque sea en un solo caracter, con la pregunta almacenada en el registro, el computador indicará dato inexistente.

Opción: 4) "Ejercitación"
En esta opción la computadora elegirá una pregunta al azar y pedirá que se ingrese la respuesta. Una vez ingresada ésta última el computador indicará si es correcta o no.

Opción: 5) "Listado completo"
Esta opción permite verificar todos los datos almacenados. Presionando la tecla de escape (=) el programa retorna al menú principal.

Opción: 6) "Grabar datos en cinta"
Con esta opción se pueden acumular todos los datos en cassette. El programa se graba junto con los datos eliminándose los problemas de archivo.

Opción: 7) "Fin del programa"
La opción 7 detiene la ejecución del programa. Digitando GOTO 40 se ejecuta nuevamente con los datos intactos.

TABLA DE VARIABLES

P$() Matriz que acumula todas las preguntas
R$() Matriz que acumula todas las respuestas.
N() Matriz que acumula las longitudes de las preguntas.
M() Matriz que acumula las longitudes de las respuestas.
DA Variable que cuenta los datos existentes en el registro.
K$ Variable que acumula el string que luego será transferido a su respectiva matriz o será utilizado para buscar datos en el registro.
X Variable que toma las opciones en el menú principal.
E$() Matriz que será dimensionada de acuerdo con los espacios vacios, luego será sumada al dato ingresado para poder compararlo con los datos almacenados en el registro.
A Variable que indica la posición donde debe empezar a escribir el cursor.
B Variable que cuenta los datos a ser listados por la opción 5.
Y Variable que indica si el dato buscado fue encontrado o no.
T Variable que indica desvío para buscar preguntas o respuestas.
KL Variable que indica el retorno al menú principal, si se ha presionado la tecla de escape.
TX Variable que indica la cantidad máxima de caracteres que debe tener cada dato.
X$ Variable que toma los caracteres a ser escritos por el cursor.

SUGERENCIAS

1) Para ingresar más datos, colocando una expansión de memoria, se deben modificar las siguientes líneas:

```
10 DIM P$(°, 96): DIM R$(°, 288): DIM N(°): DIM M(°)
55 PRINT INK 3; AT 17,0; "Datos usados"; DA; AT 17,17; "Datos libres"; ° — DA
1000 IF DA = ° THEN CLS: PRINT AT 10,6; "*Memoria excedida*": PAUSE 100: RETURN
```

(° = Cantidad máxima de datos)

2) Si se quiere modificar el sonido que se produce al escribir con el cursor se debe corregir la línea 8040.

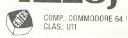
3) Para modificar la música que produce el ordenador al responder una pregunta correctamente, se debe corregir la línea 4040■



```
10 REM DATABASE
12 POKE53280,13:POKE53281,11:POKE650,128:PRINT"%":GOSUB16:IFX=0THENGOTO66
14 GOTO68
16 D$=CHR$(0):MR$=D$:DR$=D$:S=0:B1$=CHR$(10):PW=0:CW=0:B$=CHR$(32)
18 NC=0:PG=0:NL=0:F1=0:F2=0:F3=0:L$=D$:RL=0:SB$=D$:CR$=CHR$(13):HN$=D$:ID$=D$
20 A$=D$:C$=D$:T%=0:I$=D$:CK=0:I=0:J=0:K=0:L=0:M=0:N=0:RW=5:SF=0:Z=0:E$="EOF"
22 MEM=31690:EN=0:EM$=D$:ET=0:ES=0:A1$=D$:A2$=D$:A3$=D$:RETURN
24 DIMF$(F+1),T%(F+1),L%(F+1):RETURN
26 DIM REC$(R+1,F+1),ML$(9,4),PC(10),TT$(5),HC$(9),K%(R+1):RETURN
28 REM---GET---
30 GETA$:IFA$=""THEN30
32 RETURN
34 REM---CREACION---
36 IFCK<>0THENGOSUB394
38 PRINT"          INICIALIZANDO DATABASE            ":PRINT:PRINT
40 CLR:GOSUB16:INPUT"qCUANTOS CAMPOS EN CADA REGISTRO? 0      ";F:IFF=0THEN68
42 GOSUB24:FORI=1TOF
44 PRINT"        CAMPO NRO";I:PRINT"QTITULO ?
46 PRINT"LONGITUD ?
48 PRINT"      ";TAB(7);:INPUTF$(I):PRINT TAB(9);:INPUTL%(I):NEXTI
50 REM---DETER NRO DE REGISTROS---
52 FORJ=0TOF:RL=RL+L%(J):NEXTJ:RL=RL+3*(F+1)+5:R=INT((MEM-12*(F+1)-2100)/RL)
54 PRINT" SU SELECCION LE PERMITIRA TENER APROX"
56 PRINTR;"REGISTROS.   A CEPTA O   R EINGRESA?"
58 GOSUB30:IFA$="R"THENPRINT" ":GOTO38
60 IFA$="A"THENGOSUB26:CK=1:GOTO68
62 GOTO58
64 REM---MENU---
66 PRINT"  R              DATABASE               ":GOTO70
68 PRINT" ":PRINT"              MENU PRINCIPAL            "
70 PRINT"    C REAR UN ARCHIVO"
72 PRINT"    A GREGAR REGISTROS AL ARCHIVO"
74 PRINT"    M ODIFICAR LOS REGISTROS DEL ARCHIVO-
76 PRINT"    B ORRAR REGISTROS DEL ARCHIVO"
78 PRINT"    L EER EL ARCHIVO DESDE EL DISCO"
82 PRINT"    V ER EL ARCHIVO SOBRE PANTALLA"
84 PRINT"    O RDENAR EL ARCHIVO"
36 PRINT"    E SCRIBIR EL ARCHIVO EN DISCO"
88 PRINT"    *  DIRECTORIO DEL DISCO"
89 PRINT"    S ALIR DEL PROGRAMA"
90 PRINT"         OPRIMA LA TECLA APROPIADA      "
92 PRINT"        HAY";X;"REGISTROS EN MEMORIA"
94 IFR>0THENPRINT"      ESPACIO PARA";R-X;"REGISTROS MAS";
96 GOSUB30:IFA$="A"THENGOSUB350:GOTO124
98 IFA$="M"THENGOSUB350:GOTO243
100 IFA$="B"THENGOSUB350:GOTO272
102 IFA$="C"THEN36
104 IFA$="L"THEN170
108 IFA$="V"THENGOSUB350:GOTO192
110 IFA$="E"THENGOSUB350:GOTO144
112 IFA$="O"THENGOSUB350:GOTO304
114 IFA$="S"THEN342
118 IFA$="*"THEN 422
120 GOTO96
```

PROGRAMAS

RELOJ DIGITAL



COMP.: COMMODORE 64
CLAS.: UTI







El programa funciona en nuestro televisor como si fuera un reloj digital, dando en la imagen las horas, minutos y segundos. Además, podemos colocar el horario para que funcione la alarma.

```
10 GOSUB470
20 PRINT"[illegible]":POKE53269,255
30 PRINT"[illegible]"
40 PRINT"[illegible]"
50 PRINT"[illegible]"
60 PRINT"[illegible]"
70 PRINT"[illegible]"
80 PRINT"[illegible]"
90 PRINT"[illegible]"
100 PRINT"[illegible]TERMINAL HORARIA"
110 PRINT"[illegible]HORA DE ALARM
A:";"<";W$;">"
120 PRINT"[illegible]"
130 PRINT"[illegible]"
140 PRINT"\              ┌"
150 PRINT"\              ┌  COMMODORE SOFWA
RE"
160 PRINT"\              ┌"
170 PRINT"\CARLOS SOFT┌  1985 ARGENTINA"
180 PRINT"[illegible]"
190 GOTO360
200 POKE53269,0:POKE 53264,1
210 POKE53277,63:POKE53271,63:Y=106
220 POKE53248,32:POKE53249,Y:REM 0
230 POKE53250,240:POKE53251,Y:REM 1
240 POKE53252,184:POKE53253,Y:REM 2
250 POKE53254,136:POKE53255,Y:REM 3
260 POKE53256,80:POKE53257,Y:REM 4
270 POKE53258,32:POKE53259,Y:REM 5
280 POKE53260,50:POKE53261,200:REM 6
290 POKE53262,74:POKE53263,200:REM 7
300 FORI=53287 TO 53292:POKEI,1:NEXTI
310 POKE 2046,180:POKE 2047,181
320 POKE 53293,4:POKE53294,4
330 FORI=54272TO54296:POKEI,0:NEXTI
340 POKE54296,15:POKE54272,139:POKE54273
,100:POKE54275,8:POKE54278,240
350 GOTO20
360 A$=TI$
370 IFTI$=W$ THEN GOTO1050
380 A1$=RIGHT$(A$,1):A1=VAL(A1$):POKE204
0,170+A1:IF A1=T THEN GOTO400
390 POKE54276,33:FORI=0TO50:NEXTI:POKE54
```

```
276,32
400 T=A1
410 A1$=LEFT$(A$,5):A2$=RIGHT$(A1$,1):A2
=VAL(A2$):POKE2041,170+A2
420 A1$=LEFT$(A$,4):A2$=RIGHT$(A1$,1):A2
=VAL(A2$):POKE2042,170+A2
430 A1$=LEFT$(A$,3):A2$=RIGHT$(A1$,1):A2
=VAL(A2$):POKE2043,170+A2
440 A1$=LEFT$(A$,2):A2$=RIGHT$(A1$,1):A2
=VAL(A2$):POKE2044,170+A2
450 A2$=LEFT$(A$,1):A2=VAL(A2$):POKE2045
,170+A2
460 POKE51,20:POKE52,133:GOTO360
470 REM INISIALIZACION:
480 T=0:DIM D$(13)
490 POKE53272,28:POKE53281,0:POKE53280,0
500 PRINT"■■TERMINAL HOARIA":GOSUB1210
510 PRINT"■INGRESE LA HORA ACTUAL:"
520 PRINT"■ L●●●●●●●●●●●●●●●●┐"
530 PRINT" ╲                 ┌"
540 PRINT" ╱♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥_■":X=91
550 POKE53248,134:POKE53249,X
560 POKE53250,117:POKE53251,X
570 POKE53252,092:POKE53253,X
580 POKE53254,075:POKE53255,X
590 POKE53256,050:POKE53257,X
600 POKE53258,033:POKE53259,X
610 FORI=53287 TO 53292:POKEI,2:NEXTI
620 POKE53269,63:POKE53277,0:POKE53271,0
:POKE53264,0
630 REM HOLA
640 FORI=2040 TO 2045:POKEI,170:NEXTI:X=
T
650 X=X+1
660 PRINT"  ↑↑"
670 GETD$(X):IFD$(X)=""THEN670
680 IF D$(X)>"2" OR D$(X)<"0" THEN670
690 A1=VAL(D$(X)):POKE 2045,170+A1
700 PRINT"□     ↑↑":X=X+1
710 GETD$(X):IFD$(X)=""THEN710
720 IF D$(X)>"9" OR D$(X)<"0" THEN710
730 IF D$(X-1)>"1" AND D$(X)>"3" THEN710

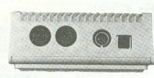
740 A2=VAL(D$(X)):POKE 2044,170+A2
750 PRINT"□         ↑↑":X=X+1
760 GETD$(X):IFD$(X)=""THEN760
770 IF D$(X)>"5" OR D$(X)<"0" THEN760
780 A3=VAL(D$(X)):POKE 2043,170+A3
790 PRINT"□            ↑↑":X=X+1
800 GETD$(X):IFD$(X)=""THEN800
810 IF D$(X)>"9" OR D$(X)<"0" THEN800
820 A4=VAL(D$(X)):POKE 2042,170+A4
830 PRINT"□                ↑↑":X=X+1
840 GETD$(X):IFD$(X)=""THEN840
850 IF D$(X)>"5" OR D$(X)<"0" THEN840
860 A5=VAL(D$(X)):POKE 2041,170+A5
870 PRINT"□                   ↑↑":X=X+1
880 GETD$(X):IFD$(X)=""THEN880
890 IF D$(X)>"9" OR D$(X)<"0" THEN880
900 A6=VAL(D$(X)):POKE 2040,170+A6
910 PRINT"□ CORRECTO (S/N)?□":GOSUB990
920 GET D$:IFD$="" THEN920
930 IF D$="S" THEN PRINT"
  ":GOTO960
940 IF D$="N" THEN PRINT"
 □":GOTO640
950 GOTO920
960 E$=D$(1)+D$(2)+D$(3)+D$(4)+D$(5)+D$(
6):TI$=E$
970 PRINT"■■■■INGRESE LA HORA: DE LA ALAR
MA■■■■■"
980 FORI=2040 TO 2045:POKEI,170:NEXTI:GO
TO650
990 IF X=6 THEN X=7:RETURN
1000 GETD$:IF D$="" THEN1000
1010 IF D$="S" THEN GOTO1040
1020 IF D$="N" THEN X=7:PRINT"
     □":GOTO650
1030 GOTO1000
1040 E$=D$(8)+D$(9)+D$(10)+D$(11)+D$(12)
+D$(13):W$=E$:G=2:GOTO200
1050 PRINT"■■■■■■■■■■■■■■■■■-F1-□ STOP
 ■ -F3-□ CONTINUAR"
1060 FORI=54272 TO 54296:POKEI,0:NEXTI
1070 POKE54296,15
1080 POKE54272,139:POKE54273,90
1090 POKE54279,139:POKE54280,150
1100 POKE54274,0:POKE54275,8
1110 POKE54281,0:POKE54282,8
1120 POKE54277,000:POKE54278,240
1130 POKE54284,000:POKE54285,240
1140 POKE 54276,33:POKE54283,65
1150 FORI=0 TO 50:NEXTI
1160 POKE54276,32:POKE54283,64
1170 GETY$
1180 IF Y$="■" THEN T=7:GOTO490
1190 IF Y$="■" THEN GOTO200
1200 GOTO1140
1210 IFG=2THEN PRINT"■■■■INGRESE LA HORA
 DE LA ALARMA":GOTO520
1220 RETURN
1230 REM FIN DEL PROGRAMA
READY.
```

EL TRANSBORDADOR ATARI

520 ST ¿CIENCIA O FICCION?

Cuando se cree que todo está ya creado, cosas como éstas nos hacen dar cuenta de que la informática recién comienza. Peguémonos contra la pared más cercana, y preparémonos para ver el desfile de la 520 ST.





Invitados exclusivamente, comenzamos un viaje fantástico por ATARI ST WORLD. Un mundo espacial, cálido y poblado por infinitas informaciones.

Un mundo separado, por muy pocos límites, en dos grandes campos o universos concéntricos, el de la 520 ST y el de la 1040 ST. Universos paralelos.

En la profunda oscuridad de estos mundos, lo único que rompe deslumbrantemente esa negrura, son las más variadas y abundantes representaciones de información.

Nos trasladamos a través de un mensaje azul, que nos habla de los 512 Kbytes de RAM que posee esta primera máquina, de los 192 Kbytes de ROM que incluye, de sus cartuchos externos de 128 Kbytes y de su tamaño: 470 por 239 milímetros (largo y ancho) y 62 milímetros en su parte más alta (la trasera).

Dejamos atrás el azul mensaje, y una pequeña pero intensa fuente de información nos eleva. Su fuerza casi no nos deja leer su contenido. Hablaba de su arquitectura interna.

Nos hizo saber que poseía nada menos que un microprocesador de 16 bits, el 68000 de Motorola trabajando a una velocidad de 8 MegaHerz (¡8 millones de estados o ciclos por segundo!). Que este microprocesador, del que hablaremos en próximos números, posee 16 bits de salida de datos, 32 bits en manejo interno, y 24 bits de direcciones. Que sus registros o variables internas eran 8 y su capacidad era de 32 bits, pudiendo almacenar en cada uno de ellos números entre 0 y 4.294.967.295.

En letras más pequeñas decía que posee niveles de interrupciones, tan solo 56 instrucciones, 14 modos de direccionamiento, 5 tipos de datos, reloj de tiempo real estandar y que es sencillamente insuperable. No mentía.

Inmediatamente después de que nuestras neuronas quedaron impregnadas con esa información, y sin restablecernos todavía del mágico asombro, un paralepípedo brillante y alargado nos invitó a que lo penetráramos. Sin rehusarnos leímos su mensaje preliminar:

MEMORIA EXTERNA

Rápidamente sentimos el calor de la información. Rezaba así: *En nuestro espacio, el de la 520 ST, podrás encontrar tres tipos de drives distintos: SF 314, SF 354 y SH 204.*

El primero te ofrece la posibilidad de almacenar 720 Kbytes de información en discos de 3,5 pulgadas de ambos lados y a doble densidad con una velocidad de transferencia de 250 Kbits por segundo (30 Kbytes).

El segundo te permite almacenar de un solo lado 360 Kbytes, con las mismas características del anterior. Este se ha diseñado para los que necesitan otro drive pero no otros 720 Kbytes. Todo está previsto.

Y el último, es un Hard Disk o disco rígido, como el de las PC, que puede almacenar 20.000 Kbytes o 20 Megabytes. Su velocidad de transferencia es de 1.33 Megabytes por segundo.

Por supuesto la 520 ST posee en su parte trasera conectores para todos estos periféricos, así que no es necesario adquirir ninguna interfaz extraña, que generalmente no funcionan correctamente. Uselos.

Una música digna del reino espacial más profundo, nos llevó hacia una inmensa y colorida pantalla que, girando sobre sí, no nos permitía ver nada más.

Su etiqueta:

GRAFICOS Y SONIDO

Decía:



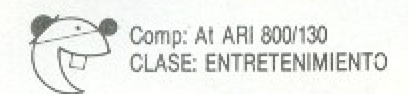
No sabemos lo que esperan de la 520, pero ella sólo puede ofrecer en color una resolución de pantalla 640 por 200 pixels con 4 colores elegidos como siempre de un set de 512 colores distintos. También en color puede ofrecer una resolución de 320 por 200 pixels con 16 colores elegidos del mismo set.
Con un monitor blanco y negro, la resolución que se obtiene es de 640 por 400 pixels, nada más.
Con respecto al sonido esta máquina le ofrece 3 canales en los cuales pueden sonar notas con frecuencias entre 30 Hz y 125 KHz. Pueden además ser afectadas por distintas formas de onda, causando los efectos de Attack, Sustain, Release, etc.
Puede además emular a una terminal VT52 via Medem Port.
Nuestra fascinación nos mantenía ya al borde del colapso informático, cuando una inmensa esfera de cristal líquido nos atrapó en su interior.
En su centro hielo tibio, tallado con laser, decía:



LO DEMAS

Y la gran esfera comenzó a derretirse, impregnándonos de la siguiente información:
Teclado: 95 teclas. 10 de función.
Microprocesador para manejar el teclado: 6301
Teclado numérico separado al igual que las teclas de cursor.
Incluye Mouse con dos botones de control y esfera removible para limpieza.
Auto-repetición de teclas.
Salida para impresora paralelo estandar.
Salida RS 232-C estandar para modem, cuyas velocidades de transmisión puede oscilar entre 50 y 19200 baudios.
Salida y entrada MIDI (Musical Instrument Digital Interface) aislado con receptores ópticos, cuya velocidad de trabajo es de 31,25 Kbaudios.
Dos pórticos destinados al Joystick o Mouse.
Sistema operativo TOS con el medio operático GEM.
Sistema multi-ventana, de tamaños cambiables o borrables utilizando la interfase virtual GEM, entre el usuario y la máquina.
Software de base: TOS (sistema operativo), GEM de escritorio, ST BASIC (lenguaje intérprete), ST LOGO (lenguaje intérprete), emulación de terminal VT 52 via modem, Panel de control para sistema de clientes, control de la configuración RS 232 y control de la configuración de impresora.
Cuando terminó de derretirse la esfera, una cautivante voz femenina proveniente de un sintetizador coloreó el espacio con:
Este es el comienzo de nuestro ínfimo universo. Como el gran universo estelar, el nuestro está en expansión...

VALIANT



Comp: At ARI 800/130
CLASE: ENTRETENIMIENTO





Un pequeño hombrecito deberá encontrar diversos objetos en los laberintos.
El primer laberinto es un bosque cerrado de pinos, en el que hallaremos una llave que nos permitirá atravesar la salida azul que se encuentra cerca de donde comenzamos a caminar.
El segundo laberinto es de ladrillos, y nos mostrará la salida al comienzo de esta etapa. Aquí los objetos factibles de ser recogidos van desde la original llave hasta espadas. Pero las cosas se complican un poco (bastante), pues, invisibles a nuestro ojos se ocultan ciertos ogros que obstruyen nuestro paso.
Esto no sería tan grave, si no fuera porque su piel es extremadamente asquerosa y letal, y con ella consumen nuestra energía vital.
El programa es muy simple, ya que todo lo que puede resultarnos dificultoso se encuentra encerrado en DATAS's.
Su lógica elemental es sencilla, y dada su corta longitud, veremos cuán buena es la calidad del mismo.

```
1 REM VALIANT
2 REM PROGRAMA PARA K 64
3 REM ATARI 400/800/600 XL/800 XL/130 XE
7 GOTO 1000
10 A=USR(1536,ADR(S$)+70*Y+X)
20 S=STICK(0):IF S=15 THEN.20
30 XD=(S=7)-(S=11)
40 YD=(S=13)-(S=14)
50 IF PEEK(40405+XD+20*YD)=0 THEN X=X+XD:Y=Y+YD:SOUND 0,100,10,10:SOUND 0,0,0,0:
GOTO 10
60 P=PEEK(40405+XD+20*YD):IF P=33 THEN 10
70 IF P=138 THEN KEY=KEY+1:EX=EX+5:GOSUB 1110
80 IF P=11 AND X=0 AND Y=40 AND SW<1 THEN 10
90 IF P=11 AND X=0 AND Y=40 AND SW=1 AND KEY>0 THEN 1470
100 IF P=11 AND KEY THEN GOSUB 1190:GOSUB 1110:GOSUB 1150:IF F THEN F=0:GOTO 135
0
110 IF P=131 THEN GOSUB 1110:GP=GP+25:EX=EX+10
120 IF P=64 THEN POSITION 5+XD,4+YD:? #6;"":GOTO 1380
130 IF P=133 THEN SW=1:GOSUB 1110
140 GOSUB 1150:GOTO 10
1000 CLR :GRAPHICS 17
1010 CB=152:POKE 756,CB:ML=1536
1020 A=USR(ADR("h"=`=ah`Po`"))
1030 DIM T$(71),S$(3710),K$(16),F$(150)
1040 READ A:IF A>-1 THEN POKE ML+K,A:K=K+1:GOTO 1040
1050 FOR J=0 TO 103:READ A:POKE CB*256+J,A:NEXT J
1060 REM
1070 GOSUB 1300
1080 GRAPHICS 17:K$="********":POKE 709,70:POKE 708,138:POKE 710,26:POKE 711,200
1090 POKE 756,CB:POSITION 12,2:? #6;"(:":POSITION 12,4:? #6;"):":POSITION 12,6:?
 #6;"':":GOSUB 1150
1100 GOTO 1210
1110 D=XD+20*YD:POKE 40405+D,0
1120 LOC=70*(Y+YD)+X+XD+285:S$(LOC,LOC)=""
1130 FOR J=100 TO 80 STEP -0.5:SOUND 0,J,10,12:NEXT J:SOUND 0,0,0,0
1140 RETURN
1150 IF KEY THEN POSITION 1,14:? #6;K$(1,KEY*2):GOTO 1160
1155 POSITION 1,14:? #6;"          "
1160 IF SW THEN POSITION 3,16:? #6;"x"
1170 POSITION 14,2:? #6;ST;"  ":POSITION 14,4:? #6;EX:POSITION 14,6:? #6;GP;"
"
```

```
1180 RETURN
1190 KEY=KEY-1:POKE 40405+XD+20*YD,0:FOR J=90 TO 100 STEP 0.5:SOUND 0,J,10,12:NE
XT J:SOUND 0.0,0,0
1200 RETURN
1210 X=10:Y=30:ST=100:F=1:A=USR(1536,ADR(S$)+70*Y+X):EX=0:GP=0
1220 POSITION 3,22:? #6;"* # valiant  %":POSITION 2,19:? #6;"OPRIMA  EL BOTON"
1230 POSITION 0,10:? #6;"* = llave# = oro":? #6:? #6;" = ogro% = espada";
1240 ? #6:? #6;"+ = puerta cerrada":? #6:? #6:"    EXIT= salida"
1250 IF STRIG(0) THEN 1250
1260 POSITION 2,19:? #6;"  BUENA SUERTE   "
1270 FOR J=-40 TO 40:SOUND 0,ABS(J)+120,10,12:NEXT J:SOUND 0,0,0,0:POSITION 0,10
:FOR J=1 TO 4
1280 ? #6;"                           ";:? #6:NEXT J:POSITION 4,19:? #6;"
        "
1290 GOTO 10
1300 RESTORE 5690:READ F$,T$:F$(LEN(F$)+1)=T$:READ T$:FOR J=1 TO 4:S$(LEN(S$)+1)
=T$:NEXT J
1310 FOR J=1 TO 44:READ A:S$(LEN(S$)+1)="LLLL":S$(LEN(S$)+1)=F$(A,A+62):S$(LEN(S
$)+1)="LLLL":NEXT J
1320 FOR J=1 TO 4:S$(LEN(S$)+1)=T$:NEXT J
1330 S$(440,440)="":S$(2589,2589)=""
1340 RETURN
1350 RESTORE 5200:S$="":FOR J=1 TO 49:READ T$:S$(LEN(S$)+1)=T$:NEXT J
1360 X=3:Y=39
1370 GOTO 10
1380 S1=STICK(0):IF S1<>15 AND S1<>S THEN POSITION 5+XD,4+YD:? #6;" ":S=S1:GOTO
30
1390 SOUND 2,10,10,12:FOR T=1 TO 4:NEXT T:SOUND 2,0,0,0
1400 ST=ST-1:GOSUB 1170:IF ST>1 THEN 1380
1410 FOR T=1 TO 500:NEXT T:? #6;"}":POSITION 1,2:? #6;"lastima  has sido"
1420 FOR J=1 TO 4:FOR K2=-3 TO 3:K=K2
1430 IF J=4 AND K2=3 THEN K=1
1440 POSITION ABS(K),5:? #6;"  V e N c I d O "
1450 SOUND 0,100-(10*J+5*ABS(K)),10,10:FOR T=1 TO 7+3*J:NEXT T:NEXT K2:NEXT J
1460 SOUND 0,0,0,0:GOTO 1490
1470 ? #6;"}":POSITION 2,2:? #6;"FELICITACIONES"
1480 POSITION 6,4:? #6;"ganaste"
1490 SCORE=EX+10*ST:POSITION 1,9:? #6;"PUNTAJE FINAL ";SCORE
1500 FOR J=10 TO 20:FOR K=1 TO 9:SOUND 0,J+K*3,10,12:FOR T=1 TO 4:NEXT T:SOUND 0
,0,0,0:NEXT K:NEXT J
1510 FOR T=1 TO 1000:NEXT T:RUN
4999 REM MACHINE LANGUAGE DATA
5000 DATA 104,104,133,204,104,133,203
5010 DATA 162,0,160,0
5020 DATA 177,203,157,129,157,232
5030 DATA 200,192,9
5040 DATA 208,245,138,24,105,11,170
5050 DATA 224,180,240,20,234
5060 DATA 165,203,24,233,185,144,2
5070 DATA 230,204,165,203,24,105,70
5080 DATA 133,203,24,144,214,169,2
5090 DATA 141,213,157,169,0,133,77,96,-1
5100 DATA 0,0,0,0,0,0,0,0
```

```
5110 DATA 251,251,251,0,191,191,191,0,24,0,60,90,0,24,36,102
5120 DATA 0,15,17,35,69,250,140,248
5130 DATA 248,30,55,51,3,6,12
5140 DATA 0,1,2,4,136,208,96,112,152
5150 DATA 219,90,66,60,24,24,36,102
5160 DATA 224,128,160,174,234,14,8,8
5170 DATA 224,128,224,46,228,4,4,4
5180 DATA 224,128,192,138,234,4,10,10
5190 REM DATA 0,6,6,6,6,102,60,0
5191 DATA 224,160,224,16,8,12,22,4
5192 DATA 255,60,60,44,60,60,60,255
5193 DATA 8,28,42,28,42,73,8,0
5200 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5210 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5220 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5230 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5240 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAA@AAAAAAAAA
5250 DATA AAAAAAAAAAAAAAAAAAAAA
5260 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAA@@@AAAAAAAAAA
5270 DATA AAAAAAAAAAAAAAAAAAAAA
5280 DATA AAAAAAAAAAAAAAAAAAAAA@AAAAAAAAAAA
5290 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5300 DATA AAAAA@AAAAAAAAAAAAAAAAAA
5310 DATA AAAA@AAA@@AAAAAAAAAAAAAAAAAAAAAA
5320 DATA AAAA@AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5330 DATA AAAAAAAAAAAAAAAAAAAAA
5340 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5350 DATA AAAA@@A@AAAAAAAAAAAAAAAAAAAAAAAAA
5360 DATA AAAAA@@AAAAAA@AAAAAAAAAAAAAAAAAA
5370 DATA AAAA@@AAAAAAAAAAAAAAAAAAAAAAAAAAA
5380 DATA AAAAA@AAAAAAAAAAAAAAAAAAAAAA
5390 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5400 DATA AAAAAAAAAAAAAAAAAAAAAAAAA
5410 DATA AAAA@@AAAAAAAAAAAAAAAAAAAAAAAAAA
5420 DATA AAAA@A@AAAAAAAA@AAAAAAAAA
5430 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5440 DATA AAAAA@AAAAAAAAAAAAA
5450 DATA AAAA@AA@A@AAAAAAAAAAAA
5460 DATA AAAAA@A@A@AAAAA
5470 DATA AAAAAAAAAA@AAAAA@@AAAA
5480 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAA
5490 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5500 DATA AAAA@AAAAAAAAAAAAAAAA
5510 DATA AAAAAAAAAAAAAA@AA@AAAAAAAA
5520 DATA AAAAAAA@AAAA@@AAAA
5530 DATA AAAA@AAAAAAAA@@AAAAAAAAA
5540 DATA AAAAAAAAAAAAAAAAAAAAAA
5550 DATA AAAA@AAAAAAAAAAAAAAA
5560 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5570 DATA AAAA@AAAAAAAAAAAAAA
5580 DATA AAAA@AAAA@@@AAAAAAAAAAAAA
5590 DATA AAAA@A@AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5600 DATA AAAA@AAAAAAAAAA
5610 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5620 DATA AAAAAAAAAAAAAA@AAAAA
5630 DATA AAAA@@A@AAAAAAAAAAAAAAAAAAAAA
5640 DATA AAAAAA@AAAAAAA@AAAAA
5650 DATA AAAAA%8)4AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5660 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5670 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5680 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5690 DATA LLLLLLLLLLLLLLLLLLLLLL
5700 DATA LLLLLLLLLLLLLLLLLLLLLLLL
5710 DATA LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
5720 DATA 10,9,4,21,12,14,11,9,5,2,19,29,5,19,21,32,16,7,18,19,22,2,6,3,5,7,4,2,
1,4,7,4,12,16,12,21,30,6,3
5730 DATA 9,3,1,4,8,12,16,12,7,2,15,20,22,25,28,34,19,8,4
```

Para su

COMMODORE 128

TODO EN UN SOLO DISKETTE
FACIL DE OPERAR
MAS RAPIDO QUE UN PC

Sistema de gestión comercial

FACTURACION: Hasta 2000 comprobantes mensuales. Actualiza stock y ctas. ctes.

CUENTAS CORRIENTES: Hasta 2000 cuentas. Resúmenes de cuentas.

CONTROL DE STOCK: Hasta 3500 artículos.

LIQUIDACION DE IMPUESTOS: I.V.A. Convenio Multilateral.

ESTADISTICA DE VENTAS: Sepa qué vendió y cuanto.

LIQUIDACION DE COMISIONES: Hasta 46 vendedores.

Y también:
Contabilidad general. Historias Clínicas médicas. Administración de consorcios. Sueldos y jornales. Control de Stock. Gestión financiera (hasta 4500 cheques diferidos). Lenguajes de programación. Bases de datos. Procesadores de textos. Hojas electrónicas. Programas a medida.



Rivadavia 5040 Local 21 – Capital Federal
Tel. 431-1081

Envíos al interior del país contra reembolso

PROGRAMACION PASCAL

El problema que por norma general se plantea primero cuando se utiliza un lenguaje compilado es aprender a abordar el proceso de múltiples etapas requerido para obtener siquiera un pequeño programa que sea ejecutable. Por: Hugo D. Caro



Para empezar se debe ingresar el texto fuente a través de algún editor de texto o procesador de texto.
Luego, tras haber almacenado el **fuente** en cinta o disco, se debe cargar el compilador del lenguaje e indicarle que compile el fuente a alguna forma de código de máquina (a menudo con una compleja cadena de opciones de líneas de comando).
Por último, este archivo "objeto" se debe "linkeditar", o sea, enlazar con las rutinas de biblioteca "runtime" necesarias. Posiblemente el programa se pueda cargar y ejecutar sin ningún otro esfuerzo, pero si el compilador genera un "seudocódigo" o código intermedio, se deberá utilizar un intérprete "runtime".
Si todo esto suena demasiado complicado, nos tranquilizará el hecho de que todos los paquetes de PASCAL disponibles para las computadoras MSX evitan en gran medida estos "trámites burocráticos". Al menos durante el desarrollo del programa, su texto fuente, compilador y programa objeto pueden estar residentes en memoria, tanto para el Hi-Soft Pascal (cassette) o el Turbo-Pascal (diskette).
Esto es posible debido a la eficiencia y el pequeño tamaño del Pascal, y el sistema resultante con frecuencia no es más complicado, al utilizarlo, que un sistema BASIC residente.
Cada sistema posee su propio juego de instrucciones para controlar el editor y el compilador, y deberemos remitirnos a menudo a los manuales correspondientes. Con mucha frecuencia, todo cuanto se necesita es una simple E para editar, C para compilar y R para ejecutar (es el caso de HiSoft y el Turbo Pascal). Lo que en este momento vamos a estudiar es la sintaxis correcta para ingresar un programa, independientemente de lo sencillo o complejo que éste pueda ser. Por suerte, el PASCAL está tan estandarizado que casi no serán necesarias aclaraciones indicando las diferencias entre las distintas versiones (si bien, cuando ingresemos en una etapa posterior de este curso, se deberán indicar algunas ampliaciones y/o diferencias del Turbo-Pascal, principalmente). Bueno, manos a la obra: veremos nuestro primer programa en PASCAL:

```
PROGRAM PRIMERO(Output);
Const
     Mensaje='Programacion Pascal K-64';

Begin
     Write(Mensaje)
End.
```

Antes de estudiar a fondo este ejemplo, probaremos ingresarlo en nuestro compilador y ejecutarlo. Por ejemplo, si tenemos el HiSoft-Pascal deberemos hacer lo siguiente:
Carguemos el Hisoft-Pascal con BLOAD"CAS:", R
Pulsemos RETURN a las preguntas sobre memoria, y una vez en el menú principal, pulsemos E para editar. Ingresemos el texto. Tengamos cuidado con el punto y coma tras la primera y tercera línea, y el punto final del programa. Una vez ingresado el programa, pulsemos CTRL + STOP para salir. Pulsemos C para compilar. Si obtenemos alguna queja por parte del compilador, leamos atentamente el mensaje de error y veamos si podemos detectar qué es lo que está mal. Luego pulsemos R para ejecutar.
Para Turbo-Pascal:
Con el indicador del sistema, pulsemos:
A TURBO
Luego pulsemos Y para indicar que deseamos los mensajes de error. Pulsemos E para editar el texto del programa. Luego pulsemos CTRL , K , D para salir del editor. Ahora pulsemos R y deberemos ver por pantalla lo siguiente (al igual que con el Hisoft-Pascal): `Programa cion Pascal K-64`
El programa es trivial, pero demuestra la forma general que tendrá cada módulo (programa, procedimiento o función). Vemos que se tienen tres partes separadas:

1. El encabezamiento, en este caso un encabezamiento de programa.

2. Declaraciones y definiciones, en nuestro caso, la única declaración que se hace es de constante.

3. El "cuerpo" del programa, que contiene todas las sentencias ejecutables.



Las exigencias de sintaxis del Pascal, al menos para los fundamentos del lenguajes, se pueden definir más claramente utilizando los denominados "diagramas de sintaxis". Éstos son como el mapa de rutas de mano única. La ruta autorizada avanza desde arriba a la izquierda hacia abajo a la derecha, y todos los casilleros que atravesemos son o bien una "entidad sintáctica" (es decir, que se representa a sí misma) contenida en un rectángulo de bordes redondeados, o bien otro elemento descrito en alguna otra parte mediante otro diagrama de sintaxis, indicado por el casillero rectangular que lo contiene. Veamos el diagrama del cuerpo del programa Pascal clásico:

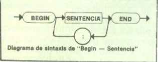


Diagrama de sintaxis de "Begin — Sentencia"

Como se puede apreciar, las palabras Begin y End están definidas como integrantes del vocabulario del Pascal, y para elaborar su significado no se requiere ningún otro diagrama. De hecho, el PASCAL reconoce sólo 35 palabras con un significado fijo, cuya lista completa ofrecemos a continuación:

And	Array	Begin
Do	Downto	Else
Function	Goto	If
Nil	Not	Of
Program	Record	Repeat
Type	Until	Var
Case	Const	Div
End	File	For
In	Label	Mod
Or	Packed	Procedure
Set	Then	To
While	With	

Nuestro programa contiene sólo cuatro de estas palabras reservadas, como también se las suele denominar; Program, Const (de *constant*: constante), Begin y End. La palabra Program es un "identificador" que identifica el nombre del programa y puede ser cualquier identificador legal que se elija. En el caso de esta sentencia es opcional para los compiladores que están disponibles en MSX, aunque nosotros los incorporaremos para mayor claridad y para mantenernos dentro del Pascal standard.

El nombre del programa debe ser un identificador válido, y debe comenzar con una letra o número. Vemos que, comenzando con una letra y cambiando solamente letras y números, la cantidad de nombres que podríamos utilizar es inmensa. Sin embargo, es lógico que esté restringido el uso de nombres a aquellos que NO sean palabras reservadas. Por ejemplo,

```
Nombre
TURBO
PrograMSX
N
ABC123
CodigoPostal
Nombreyapellido
IdentificadorReal
menteKilometrico
```

son todos legales, en cambio,

```
Prog-1
UNOyMEDIO%
Consumo.Poblacion
and
Temporal$
No-se-dio
1024-Codigo
Es logico que
123Fiado
Si sucede que
```

son ilegales, ya sea porque poseen símbolos que no son alfanuméricos, o porque empiezan por un número o (en el caso de and) son alguna de las palabras reservadas del PASCAL. El último ejemplo es ilegal porque se utiliza un espacio para separar palabras, siendo legales cada palabra componente (Si, sucede y que). En Pascal, como en castellano o inglés, las letras en minúsculas o mayúsculas no representan ninguna diferencia de significado, si bien algunas versiones no estandarizadas exigen que las palabras reservadas vayan en mayúsculas.

En lo visto hasta el momento, ya sabemos lo que es válido y no con respecto a nombres de constantes y variables. Pero notamos en nuestro pequeño programa que se puede diferenciar una sección respecto del programa principal (que comienza con begin); las declaraciones. En Pascal se deben *declarar* las variables y constantes antes de utilizarlas, ya que el compilador no reserva por sí mismo el espacio de memoria para cada variable. Existen diversas razones para utilizar declaraciones de variables, de las cuales podremos analizar las siguientes:

1. Una observación del conjunto

de variables, estando todas ellas agrupadas, hace más fácil al lector la comprensión de la finalidad del programa. Se mejora aún más esta característica utilizando nombres de variables con significado (por ejemplo tasa en lugar de r) y agregando comentario para explicar el uso de las variables. Una disposición tal del programador es una de las primeras recetas del manual del buen programador.

2. Si nos detenemos a pensar en la sección de declaración de variables y constantes, inevitablemente deberemos realizar una cierta planificación del programa antes de comenzar a escribirlo. Por ejemplo, ¿con qué información se puede ejecutar del programa? ¿Qué es lo que deseo exactamente que imprima?

3. La declaración de variables ayuda a prevenir uno de los errores de programación más sutiles y difíciles de encontrar: el cambio accidental de una letra en el nombre de la variable. Por ejemplo, supongamos que en un determinado lenguaje, cuyo nombre nos reservamos, escribimos la sentencia: `LOMO=430.00`

y, durante el programa, introducimos equivocadamente:

```
PRECIO = 0.150 * L0MO
 - 20.0
```

en donde accidentalmente hemos sustituído la letra O por el número 0. El programa creará una nueva variable llamada L0MO, y utilizará cualquier valor que se le ocurra a ella (quizá cero, quizá basura). Por lo tanto, PRECIO tendrá un valor equivocado, y llevará un tiempo respetable detectar qué ha sucedido en realidad. Esto no puede suceder en Pascal (a menos que se sea lo suficientemente tonto como para declarar dos variables con un aspecto tan semejante), ya que el compilador se encargará de avisar que la variable L0MO no está declarada.

4. Nuestro programa en Pascal no funcionará, a menos que declaremos las variables.

Además de variables, hemos visto que también se pueden declarar constantes. Existe una sutil diferencia entre la declaración de variables y constantes:

La declaración de variables reserva áreas de memoria para su posterior uso, como "casilleros" donde se irán guardando los datos procesados. En cambio, la declaración de una constante es, más bien, una directiva al compilador, indicándole que cada vez que nos refiramos a la constante, en realidad queremos decir su equivalente. En nuestro programa, la instrucción:

```
write(Mensaje)
```

equivale a:

```
write('Programacion
Pascal K-64')
```

Por otra parte, las constantes no pueden variar durante la ejecución del programa (por ello son "constantes"), mientras que las variables, sí (ya que son "variables").

Aparte de los espacios y el final de una línea, en la sintaxis del Pascal hay otro elemento que se puede emplear como separador: un comentario. Éste puede aparecer en cualquier lugar del texto, excepto, por supuesto, en medio de las palabras. Los comentarios se delimitan mediante "llaves"({}).

Arriesguémonos un poco más, e ingresemos algo que se parezca más a un auténtico programa en Pascal:

```
Program ProgramDos
    (input,output);
{ Calcula el cuadrado
     de un numero }

Const
     Pedido = 'Ingrese
un numero';

Var
     numero : integer;

Begin
     Writeln;
     Writeln;
     write(Pedido);
     read(numero);
     WriteLn(numero;'
al cuadrado
es ',numero*numero)
End.
```

Nótese que ahora incluímos en el encabezamiento el identificador input. El Pascal requiere input y output, que identifican los archivos externos con los cuales se comunicará el programa. En nuestra MSX, normalmente serán el teclado y la pantalla, respectivamente. Recordemos que el comando Program es opcional para las versiones de Pascal para MSX, así que input y output también lo son.

Al incluir estas declaraciones, le indicamos a Pascal que lea desde teclado cada vez que ejecutemos la sentencia standard read. Al igual que write, todos los "parámetros se deben listar entre paréntesis.

La memoria utilizada para almacenar estos parámetros queda reservada por la declaración Var, en este caso para un único número entero. A diferencia del BASIC, que por lo general sólo puede distinguir entre números y datos alfanuméricos (mediante la utilización del signo dólar después del indentificador), la gama de tipos de datos de que dispone el Pascal es casi ilimitada. Por consiguiente, es importante informar al compilador sobre cuánta memoria debe reservar para almacenar cada dato, como ya se ha dicho anteriormente. Las variables simples, por lo tanto, se declaran dentro de la sentencia Var.

Sin embargo, para más comodidad, Pascal admite declaración de variables de tipo standard, dentro de la siguiente lista:

```
char        :  tipo caracter (una
    única   letra, número o
               símbolo)
integer     : tipo entero
               (como BASIC).
real        : tipo punto flotante
boolean     : tipo Verdad o Falso
```

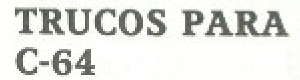
Siguiendo con el programa, veamos que sucede cuando lo corremos.

El cursor permanecerá posicionado inmediatamente después del Pedido, igual que si hubiéramos empleado una sentencia PRINT en Basic con un punto y coma final. Esto es exactamente lo que queríamos al utilizar el procesamiento incorporado write del Pascal. Siempre que se requiera una nueva línea, debemos emplear el procedimiento alternativo WriteLn. La Ln es una contracción de la palabra Line (línea), y es útil utilizar una W y L mayúsculas de cada una de las palabras que lo componen. Una sentencia WriteLn sólo creará una nueva línea■



C U R S O S

TRUCOS, TRAMPAS Y HALLAZGOS



TRUCOS PARA C-64

Si nuestro deseo es generar números aleatorios desde un programa Assembler en un rango definido, o si es llenar un bloque de memoria con un valor determinado como, por ejemplo, el destinado a la pantalla de texto, entonces, afilemos el lápiz y continuemos leyendo.

En primer lugar, la rutina de la figura 1, que como verán está hecha en Basic, se encargará como dijimos antes, de llenar la memoria de pantalla con un valor determinado y específico.

Por ejemplo, podremos llenar la pantalla con el valor 1, con lo que visualizaremos en el medio de representación catódica (el tubo), toda una trama de letras A mayúscula.

También, en el ejemplo del programa Basic, podremos especificar un color determinado, como la combinación de frente y fondo.

Este programa además se encargará de pasar a la memoria el programa Assembler contenido en los DATAs de las líneas 95 y 97.

Recordemos que no es necesario que el listado Basic permanezca en memoria una vez que se haya ejecutado por lo menos una vez.

Simplemente podremos ejecutar el programa Assembler, por medio de comando SYS 49152.

Si por error hemos copiado mal alguno de los números de las líneas 95 o 97, el mismo listado Basic se encargará de avisarnos a su tiempo, evitandonos la caída del sistema.

En la figura 3, los más interesados en el funcionamiento de esta rutira podrán saciar su curiosidad rápidamente. Este es el desensamble de la rutina que presentamos.

En la figura 2 podemos ver otro listado Basic que a su vez contiene y es el encargado de pasar a la memoria, un programa Assembler.

Una vez ejecutado el mismo, con sólo darle entrada al comando SYS50140,N, donde n será el número máximo que queremos que genere en forma aleatoria, obtendremos el resultado en la posición de memoria 251.

Para conocer el resultado de esta rutina deberemos efectuar el mando ? PEEK(251).

Así por ejemplo para obtener un número aleatorio entre 0 y 20 efectuaremos el comando SYS 50140, 20 y lo leeremos con ? PEEK(251).

Seguramente, será de gran utilidad para los programadores que todavía no incursionan en el submundo Assembler y para los que ya están bastante avanzados. Que lo disfruten...

Figura 3

DIR. HEXAD.	MNEMONICO	
C000 A0 00	LDY $00	:Carga 0 en Y
C002 A2 00	LDX $00	: Carga 0 en X
C004 A9 05	LDA $05	Carácter en A
C006 9D 00 04	STA $0400,X	:Llena 256 primeras posiciones
C009 A9 07	LDA $07	:Carga en A el color
C00B 9D 00 D8	STA $D800,X	:Llena memoria de color
C00E E8	INX	:Incrementa X
C00F D0 F3	BNE C004	:Si es 0 (256) va a C004
C011 C8	INY	:Incrementa Y
C012 EE 08 CO	INC $C008	:Incrementa C008
C015 EE OD CO	INC $C00D	:Incrementa COOD
C018 CO 04	CPY $04	:Compara Y con 4
CO1A D0 E6	BNE $C002	:no hemos llegado a a C002
C01C A9 04	LDA $04	:Carga acumulador con 4
CO1E 8D 08 CO	STA C008	:Coloca A en C008
C021 A9 D8	LDA $ D8	:Carga acumulador con D8
C023 8D 0D C0	STA $C00D	:Coloca A en COOD
C026 60	RTS	:Devuelve el control al BASIC

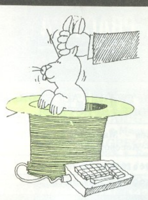
Figura 2

```
10 REM NUMEROS ALEATORIOS EN CM
20 FORI=50140TO50180
30 READA:POKEI,A:NEXT
40 DATA 32,253,174,32,138,173,32,247,183
50 DATA 201,0,208,25,132,0,240,21,162,129
60 DATA 142,18,212,162,255,142,15,212,173
70 DATA 27,212,133,251,24,196,251,144,246
80 DATA 96,76,72,178
READY.
```

Figura 1

```
10 FORI=49152TO49190:READA:C=C+A:POKEI,A:NEXT
20 IFC<>4942THENPRINT"HAY UN NUMERO EQUIVOCADO EN LOS DATA":STOP
30 INPUT "CARACTER";C
40 INPUT "COLOR";K
50 POKE 49157,C
60 POKE 49162,K:POKE53280,K
70 IF(PEEK(53281)AND15)=KTHEN40
80 SYS 49152
90 GOTO 90
95 DATA 160,0,162,0,169,4,157,0,4,169,5,157,0,216,232,208,243,200,238,8
97 DATA 192,238,13,192,192,4,208,230,169,4,141,8,192,169,216,141,13,192,96
READY.
```



VISUALIZACION DE LAS TECLAS DE FUNCION (MSX)

En los manuales de las MSX habrán encontrado una sentencia que les permite visualizar o borrar las instrucciones de las teclas de función.

Pero esa no es la única manera. En algunos programas comerciales, por ejemplo, utilizan otra manera un poco más complicada pero, según la ocasión, con mejores resultados.

Pero demos el truco de una vez, para no impacientar al estimado lector.

No, mejor lo explicaremos primero. Y ahora va la introducción: en la memoria RAM, en la dirección hexadecimal "F3DE", se almacena un 1 si se visualizan las sentencias destinadas a las funciones. Pero si no se ven, habrá en esa posición un 0.

Por lo tanto (y ahora sí viene el truco), para visualizar las sentencias definidas para las teclas de función, será:

POKE &HF3DE, 1

y, obviamente, para que desaparezcan las funciones del último renglón, deberemos entrar:

POKE &HF3DE, 0

Pero alto, aún no terminó la explicación. Según lo leído antes, las instrucciones anteriores deberían hacer desaparecer, o no, las funciones, (y por supuesto que lo hacen), pero no inmediatamente, sino que desaparecerán al entrar CLS o en el caso contrario, al presionar la tecla de SHIFT.

ASUNTO DE RUIDOS (MSX)

En las MSX, al presionar una tecla, habrán podido escuchar un chasquido (o sea, ruido muy bajo). A algunos este ruido les molesta o simplemente les arruina el programa, otros quieren ponerlo en alguna sección sólo del programa o...

Y para conformar a todos los gustos, proponemos la solución.

En la dirección de la RAM F3DB (en hexadecimal), se almacena un 1 para poder escuchar el chasquido y un 0 (en la misma posición) si se lo quiere apagar.

Entonces, y ahora viene la receta, coloquemos la sentencia:

POKE&HF3DB, 1 para escuchar las pulsaciones de las teclas y POKE &HF3DB, 0 para poder pulsar las teclas sin oír ningún otro ruido más que el golpe de nuestros dedos contra las indefensas teclas.

GUIA PRACTICA GUIA PRACTICA

GUIA PRACTICA GUIA PRACTICA

GUIA PRACTICA GUIA PRACTICA

GUIA PRACTICA GUIA PRACTICA

GUIA PRACTICA GUIA PRACTICA

Distribuidor Oficial de:

EN MORON

- CZ SPECTRUM - CZ 1500 - TK 90 - TK 85
- COMMODORE 16 - 24 - 128 - AMIGA.
- Service y todo tipo de accesorios
- Novedades en programas para:
  COMMODORE 16 - MSX - AMIGA.
  Transformación de SPECTRUM en PLUS
- CURSOS DE COMPUTACION: para Niños - Adolescentes - Adultos, Profesionales y empresarios, Docentes y Establecimientos Educativos, grupos de hasta 12 alumnos con 2 ó 3 alumnos por equipo.

CT COMPU TAILOR S.R.L. BROWN 749 Of. 6 y 7 Morón - 628-0821





EN BELGRANO

TENEMOS TODO PARA
MSX - C/128 - SPECTRUM, TK 90 X, C 64, TK 85, TK 2000 y 2068

PRECIOS P/CASSETTE (COPIAS MAQUINA) o DISCO A 2,00

SERVICIO TECNICO ESPECIALIZADO

DISTRIBUIDORA MAYORISTA de SOFTWARE y ACCESORIOS
ENVIOS AL INTERIOR

Av. Cabildo 2230
(1428) CAP.
Gal. Las Vegas Loc. 72

785-2668
568-9611

RILEN S. A.

DISTRIBUIDORES:

MICRODIGITAL
INTERFASES - PROGRAMAS - JOYSTICKS
CASSETTES - SERVICIO TECNICO

BOLIVAR 1218
1141 BUENOS AIRES

TE.: 361-2787
3532

ENVIOS AL INTERIOR



le propone...

Ahora, para acelerar la carga en la C-128 hay otra opción:

MACH 128 *(cartridge + disco utilitario)*

Trabaja en modo 64 y 128 / multiplica varias veces la velocidad de carga / infinidad de comandos: hardcopy, etc.
Lo más novedoso: cartridge y disco totalmente en CASTELLANO!!!

...busque la nueva imagen de jdc computación

FABRICANTE DE ACCESORIOS PARA COMMODORE

OLAZABAL 5142 8° "40"
51-0021/52-3967

ATENCION: USUARIOS DE COMMODORE 64 - 128

1800 programas exclusivos en Cassettes
2300 en Diskettes - Ventas por Mayor y Menor de: Interfases - Reset - Fundas - Transformadores - Reparación de Consolas y Datassette - Manuales en Castellano. Joystick y Dataset Cintas de Impresión Formularios Contínuos

S'AGO OMEGA

SANABRIA 3208 (1417)
SABADOS ABIERTO TODO EL DIA
ENVIOS AL INTERIOR

DOS AMIGOS



1 CONSOLA 64
1 DATASSETTE
2 JOYSTICKS
1 JUEGO
1 FUENTE

HORARIO: LUN. A VIERNES 9 a 20 Hs.
Sabado Abierto

OFERTAS

1 CONSOLA 64
FUENTE Y CABLES

SERVICIO TECNICO - LAB. PROPIO
CONVERSIONES C-64/128 y ATARI PAL-N
REPARACIONES - ASESORAMIENTO INTEGRAL
SOFT 64 - 128 - CP/M

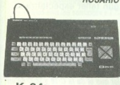
1 CONSOLA 128
1 DRIVE 1571
FUENTE ORIGINAL
CABLES
EL MEJOR PRECIO DE PLAZA

GURRUCHAGA 105 (1414)
TE.: 854-2060

MSX Y DUDAS

1.— Quisiera saber si existe algún FAST LOAD para la TALENT MSX DPC 200
2.— Quisiera saber cuál computadora es mejor, si la C-64 o la MSX y por qué
3.—¿Qué es el hard y el software?
4.— Quiero agradecerles todo lo que hacen para que nosotros estemos conformes.

SEBASTIÁN D. AGOSTINO
ROSARIO



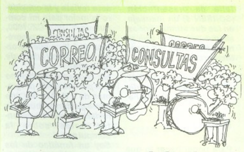
K 64:

1.— Hasta el momento no hay ningún acelerador de disco para la TALENT. El motivo de ésto, puede ser que la DPF 550 es lo suficientemente rápida como para no necesitar ningún tipo de aditamento externo.
2.— Son dos máquinas distintas, una apoyada fundamentalmente por la gran cantidad de soft (la C-64) mientras que la MSX abre un nuevo horizonte en cuanto a capacidad y compatibilidad. Todavía queda mucho por decir y hacer con respecto a la MSX, siendo esta máquina nueva en el mercado.
3.— Para contestarte en forma sencilla, el hard es la parte "dura", vale decir las computadoras, los circuitos, el medio en el cual el soft (los programas) pueden ser desarrollados.

FUTURO USUARIO

Estoy por comprarme una C-64, y he visto numerosos programas publicados en su revista para esta máqina. En ellos se utilizan a menudo las sentencias PEEK y POKE. Se que se utilizan para manejar lugares de memoria, pero no entiendo bien su funcionamiento. Me gustaría que me explicaran para qué sirven estas dos instrucciones.
Además, tengo un grabador Pionner, y quisiera saber si me va a servir para grabar los programas de la C-64 o si es conveniente comprar otro.

SILVIA LAMADRID
SAN ANTONIO DE ARECO



Esta sección está dedicada a responder las dudas que aparezcan. Les pedimos que nos escriban a nuestra casa: Paraná 720, Piso 5°, (1017) Capital Federal

K 64:

Las instrucciones PEEK y POKE se utilizan como vos bien apuntaste, para modificar ciertas posiciones de memoria. La instrucción PEEK, se utiliza para leer el valor de una posición de memoria, y su formato es PEEK n, donde n representa el lugar de memoria que estás leyendo. El resultado de esta instrucción será un número que estará en el rango de 0 a 225, y será el byte que ocupe la dirección de memoria n.
En cuanto al POKE, la función de éste es la inversa del PEEK. Mientras que el PEEK lee una dirección, el POKE almacena un valor en la dirección n. El formato de la instrucción es: POKE n, de; donde n es la dirección de memoria, y d es el dato que queremos almacenar. El valor del dato d debe estar entre 0 y 255, y será el byte que quedará guardado en la dirección n.
Lo que es más complejo de explicar, es el efecto de estas instrucciones, en especial del POKE. Dependiendo de la dirección de memoria que estemos utilizando, podemos lograr efectos que van desde colgar la máquina, hasta proteger listados o cambiar colores en la pantalla.
En cuanto al tema del grabador, para poder utilizar tu Pionner vas a tener que comprar una interfase que te permita utilizarlo como un datasette de Commodore. De otro modo, éste no te será útil, y la otra opción es comprar un datasette específico para la C-64.

GRABADOR REBELDE

Cada tanto alguna grabación sale con muy bajo nivel y he llegado a perder hasta una hora de trabajo tratando de cargar un programa. Mantengo el cabezal del grabador limpio, y el service de la TK 90 no encontró ningún defecto en la máquina ni en el grabador. ¿Qué puede estar pasando?

CARLOS HUDER
LANÚS

K 64:

Si ya descartaste un problema de máquina o de grabador, te queda probar con otro tipo de cinta. Algunas cintas de baja calidad ensucian el cabezal cuando se están grabando, mientras que otras se desmagnetizan rápidamente. Te recomendamos que pruebes primero con otro tipo de cinta, si el problema se repite probá cambiar el grabador, y finalmente cambia los cables.

Importado

Tengo una TK 90 y quisiera saber si me conviene conseguir alguna impresora o un drive en el exterior (España) por la compatibilidad con el conector trasero de la Spectrum.
¿En el país, existen impresoras de 80 columnas?
Pienso comprar un assembler, ¿cuál es el mejor?
¿Tienen información sobre un lápiz óptico para la TK 90?
¿Hay en existencia algún libro sobre el mapa de la RAM para la TK 90 o la Spectrum?

Fernando Begue
Adrogué

K 64

En cuanto a traerte algo de afuera, te diría que lo único que puede valer la pena es un drive que no puedas conseguir en el mercado local. Con respecto a la impresora, en el país hay una amplia variedad de ellas. Te sugerimos que recorras un poco las casas de computación.
Uno de los mejores assemblers para la spectrum es el Zeus, que hemos comentado y explicado en nuestra revista. En cuanto al lápiz óptico, está en venta para la TK 90, y podés averiguar su precio y características en los distribuidores de Microdigital.

El mapa de memoria de la Ram, lo podés encontrar en el manual de la máquina, en la última parte.

Más telemática

Quisiera sugerirles que den mayor cabida al tema de la Telemática. Lamentablemente, en nuestro país este importantísimo medio de aplicación de una computadora no está ni difundido ni convenientemente promocionado. Podrían tratar temas como: Sistemas de comunicación, Bases de datos, Formas de transmisión, notas sobre la situación en nuestro país (datos y direcciones de bases de datos, red ARPAC), etcétera.
Espero que hagan eco de mi propuesta, ya que creo que es común a muchos aficionados a la informática, y si no es así, es verdaderamente una pena que no haya interés en lo que pienso de una de las publicaciones más útiles, aprovechables y completas de la computación.

Hernán Polo
(5016) Córdoba

K-64:

Hacemos todo lo que está a nuestro alcance. Un ejemplo de esto son las notas "Dominando las comunicaciones" del Ing. Colla. No debés olvidar que K-64 va dirigida a todos los niveles, y no podrían saturarnos de notas técnicas sin aburrir a gran parte de nuestros lectores.

Técnicas avanzadas

La revista me parece muy buena, pero quisiera que publiquen programas para Spectrum de tipo utilitario.
Mi duda es cómo hacer para colocar en una pantalla de presentación (SCREEN $) un mensaje en las dos últimas líneas sin que se borre al salir el mensaje "Start the tape, then press any key".

Sergio Gimbernat
(2000) Rosario
Santa Fe

K-64:

Para poder hacer lo que decís, tendrías que hacer uso de la rutina de grabación no desde Basic sino desde código máquina. Esta es la única forma que se nos ocurre para poder evitar que la máquina te imprima el mensaje en la última línea. Deberías entrar a la rutina de grabación en algún punto después de que mande el mensaje a la pantalla.

Compatibilidad

Quisiera hacerles la siguiente consulta: son totalmente compatibles en software la TK 90 y la CZ 2000, y si existe en nuestro país algún disk drive para la TK 90.

Martín del Percio
Capital

K 64:

La CZ 2000 y la TK 90 no son totalmente compatibles en software, aunque son pocos los programas de Spectrum que no entran en la TK 90.
De todos modos, las pruebas que hemos realizado demuestran que no hay problema en utilizar el microdrive o la Opus Discovery en la TK 90.

Mejor con más memoria

Me llamo Gustavo Cabanillas, tengo 15 años y vivo en la provincia de Córdoba. Me gusta mucho esta revista y la considero la mejor para mi computadora, una CZ 1000, pues trae algunos programas de 2 K ya que carezco de ampliador de memoria.
Me gustaría muchísimo que pusieran más programas de 2 K. Aún así, me sigue gustando mucho la revista por sus interesantes notas sobre la computación.
Soy un fanático de los juegos electrónicos y sueño con algún día poder comprarme un ampliador de memoria para poder hacer esos buenísimos juegos como "Interceptor galáctico", "Super Olímpico" y programas educativos como el "Cometa Halley", "Constellation", etcétera.
Estoy haciendo un interesante programa de preguntas y respuestas y apenas lo termine se los mandaré para que lo califiquen.
Para mí ha sido maravilloso entrar en el mundo de la computación, porque cada vez que prendo la computadora aprendo algo nuevo y ustedes son el medio para que yo aprenda fácilmente y más rápido.

Gustavo Cabanillas
Entre Ríos 3184
(5000) Córdoba

PROBLEMAS DE HARD

Tengo un problema con mi TI 99/4A, pues cuando inserto algún módulo en la consola ésta se vuelve loca. No aparece el cartel de presentación y la máquina se queda haciendo ruidos raros. Quisiera saber ¿a qué se puede deber esto?

JUAN MARTÍN LOUSTAU
(7300) AZUL

K-64:

Pese a que no nos das demasiados datos con respecto a tu problema, deberías probar si esto se repite con todos los cartridges o con alguno en particular. En caso de que suceda con todos, debés descartar un problema del periférico y abocarte directamente a la revisación de la máquina. El defecto puede ser algo simple, como un contacto sucio en el port de cartridges. Te recomendamos que revises esto último y que limpies los contactos con algún líquido para tales efectos. Si esto no da resultado, el problema será seguramente de tipo interno y te recomendamos que lleves la máquina a algún taller especializado.

BASIC DE C 128

Quisiera hacerles dos preguntas:
1.— ¿El BASIC de la C 128 es mejor que el de la C 16 y la C 64 en cuanto a sus posibilidades gráficas?
2.— ¿La C 128 puede conectarse en sus tres modos de operación a un televisor color compatible a una C 64 y si ocurre lo mismo con la disquetera 1541 para C 64?

NESTOR A. BALICH
DOCK SUD

K 64:

El BASIC de la C 128 es sin duda mejor que el de la C 64. En cuanto a instrucciones gráficas el BASIC de la C 64 es algo pobre. En cuanto al de la C 16, si bien éste es una versión mejorada del de la C 64, dado que esta máquina no tiene Sprites, sus capacidades gráficas se ven disminuidas.
2.— En cuanto a la primera parte de tu pregunta, la respuesta es sí y no. **Sí**, siempre y cuando trabajes en el modo de 40 columnas, y **no** si pretendés trabajar en 80. En este último caso, la resolución del televisor color común no es suficiente para poder apreciar con buena definición los 80 caracteres.
En cuanto a la disquetera, podrás usarla sin problemas en modo 64, trabajará muy lentamente en modo 128 y será una tortuga en CP/M.

DUDAS DE MSX

1.— Me gustaría que me expliquen cómo y dónde conectar el modem que se expone en la página 26 de la edición de julio.
2.— ¿Cómo puedo ver los listados de los programas que se cargan con BLOAD?
3.— ¿Los desarrollos que aparecen en sus revistas para otras máquinas que tienen el Z 80 pueden servir para la mía que también lo tiene?
4.— ¿Mi máquina tiene la interfase RS 232 incorporada?
5.— ¿Qué son las ventanas y TTL?
6.— ¿Cómo puedo hacer para desenchufar parte del teclado por software?

LUIS A. QUICCHI
BAHIA BLANCA

K 64:

1.— El modem se conecta en el port trasero de la máquina y no necesita ningún tipo de interfase.
2.— Los programa que se cargan con la instrucción BLOAD están hechos en código máquina, y por lo tanto no los podés listar en forma directa. Para poder hacerlo, tendrías que tener un monitor o disassembler, y datos acerca del comienzo y finalización del mismo.
3.— Si bien dos máquinas pueden tener el mismo microprocesador, esto no quiere decir que se le puedan conectar los mismos periféricos. Esto se extiende también a los desarrollos de hard, que generalmente están concebidos para una máquina en particular y no para ser transportables.
4.— No.
5.— Las ventanas son un método de presentación de datos en pantalla que consiste en dividir a ésta en diferentes secciones llamadas ventanas que son independientes entre sí. Por "TTL" entendemos que te referís a la lógica de integración denominada así por la abreviación de Transistor Transistor Logic.
6.— Por medio de programación en código máquina, podés desarrollar tu propia rutina de lectura de teclado, y así leer sólo lo que te interesa.

QUE POKE?

Tengo una CZ 1000 y quisiera que me contesten las siguientes dudas:
1.—¿Para qué sirve la sentencia poke?
2.— ¿Se puede almacenar información en la memoria ROM? Y en este caso, ¿con qué función?
3.— En el concurso "el programador del año", ¿la elección se realiza en niveles? Pregunto esto, ya que mi máquina es pequeña y no puede competir con otras de mayor poder.

ENZO RIZZO
ENTRE RIOS

K 64:

1.— La sentencia POKE sirve para almacenar un valor determinado en alguna posición de memoria. Por ejemplo, si querés guardar el valor 56 en la dirección 20000, hacemos POKE 20000,56 y listo.

2.—En la ROM, la información se almacena en el momento de fabricarse la memoria, y luego no se altera más. La función de la misma es obvia, en ella se guarda toda aquella información que es vital para el funcionamiento de la máquina y no debe borrarse cuando se apaga la misma.
3.— Como es lógico, no podés comparar un programa hecho en una C 128 con uno hecho en una CZ 1000. El criterio de evaluación tiene en cuenta a la máquina en que fue desarrollado el programa.



CORREO • CONSULTAS

TI OLVIDADA

Poseo una TI 99 con el cartucho de BASIC extendido, y no tengo ni disquetera ni otros agregados. En un primer momento, era utilizada por toda la familia, y aprendíamos mucho de libros o revistas que se dedicaban a nuestra máquina. Desde un tiempo atrás hemos notado lo siguiente:
1.— Hace tiempo que no se consigue nueva bibliografía, entonces la máquina va a parar al ropero.
2.— En el mercado hay, aparentemente, dos opciones principales: las computadoras dedicadas a juegos, como la Commodore, y las que son versátiles como para darles algún uso comercial, caso MSX.
3.— Ahora uno piensa: ¿vendo la TI o la tiro en un cajón como recuerdo, o todavía le puedo dar alguna utilidad?
4.— Quisiera que me indiquen las diferencias entre las distintas máquinas.

JORGE ARES
MAR DEL PLATA

K 64:

En tus palabras podemos sintetizar el pensamiento de muchos poseedores de microcomputadoras, no sólo de TI, que ven como el avance tecnológico hace que sus máquinas vayan quedando olvidadas y tiendan a ser obsoletas. Esto no quiere decir que la TI sea una mala máquina, sucede siemplemente que máquinas más modernas, y los fabricantes las tienen que vender. Entonces, toda la publicidad y el esfuerzo va dirigido a estas nuevas máquinas, y uno se siente un poco olvidado, casi dejado de lado. Ante esta situación, hay dos posibilidades. La primera sería desprendernos de nuestra máquina anterior y pasar a una nueva. Pero para poder hacer esto hace falta contar con dinero, y será una historia de nunca acabar, pues todo el tiempo están apareciendo en el mercado más y mejores máquinas. La otra opción es seguir con nuestra máquina original, y si se nos acabó la bibliografía empezar a investigar por nuestra propia cuenta. También podemos recurrir al Club TI 99, avenida Pueyrredón 860, piso 9°, Capital Federal.
En cuanto a un estudio comparativo de las distintas computadoras que podés conseguir en el mercado, estamos preparando una nota al respecto que te podrá aclarar todas las dudas y será publicada en el número de diciembre.

CLUB EN FORMACION

Mi nombre es V. Margareth, tengo 18 años y soy amante de la computación. Estoy formando el "Club 1500"; para usuarios de computadoras hogareñas tipo CZ 1000, TK 81,TK83, TK 85 y compatibles. He recurrido a ustedes pues entiendo que su publicación tiene amplia difusión entre los propietarios de estas máquinas y espero de esta forma acelerar el crecimiento del club. La dirección es Montevideo 31, piso 9, dpto. "15" Capital. Desde ya les agradezco su amabilidad.

MARGARETH
CLUB 1500

Soy asiduo lector de su revista y percibo que mes a mes se superan, por lo cual les solicito publicar mi intención de formar un club de intercambio de programas para TK 85 y TK 90, de los cuales poseo más de 50 títulos. Desde ya les agradezco su atención.

DIEGO SIMUNOVICH
ALEM 1223
(2183) AREQUITO
STA. FE

K 64:

Les deseamos mucha suerte.

CHARLAS ENTRE COMPUTADORAS

Recién me inicio en el mundo de la informática, y tengo una TC 2068. Quisiera que me digan si es posible y cómo se hace para ligar una TC con una Apple II que se encuentra a distancia

SEBASTIAN CONSENTINO
TURDERA

K 64:

Para poder comunicar una computadora con otra, podés optar por varios métodos. El más fácil y standard es por medio de una interfase serie RS 232. Con el soft adecuado y un modem, podés comunicar dos computadoras distantes por medio de una línea telefónica.

MODIFICACION COMPLICADA

Quisiera saber ¿cómo se puede pasar el mando de un programa, de las teclas al joystick?
Además:
1.— ¿Qué es un mapa de memoria?
2.— ¿Qué es lenguaje de máquina?

SANTIAGO GARCIA
SALTA

K 64:

Si el programa fue escrito en BASIC, no existe mayor dificultad. Tenés que tomarte el trabajo de buscar las líneas del mismo en donde se lee el teclado, y cambiarlas por las de lectura de joystick. Desgraciadamente, no nos decís de qué máquina se trata y por lo tanto no te podemos dar el formato de estas últimas. Si el programa está en código máquina, la cosa se complica, y si aceptás un sano consejo te diríamos que las dejes así.
Un mapa de memoria es una forma de representar cómo están distribuidas las distintas secciones que componen la memoria de tu computadora, asignándole a cada una de ellas un principio y un fin. Estos límites pueden ser fijos o variables.
Lenguaje de máquina es el idioma nativo del microprocesador (el cerebro de tu computadora). Como este es bastante difícil de aprender, en especial para los que recién se inician, fueron creados los lenguajes de alto nivel (como el BASIC). Cada línea que vos escribas en BASIC será traducida a código máquina en forma interna a la computadora, para que el micro la pueda ejecutar.

NUMERO EQUIVOCADO

En el artículo que se llama "Interfase para discado telefónico automático", me quedaron algunas dudas acerca de su función.
Yo entendí dos cosas:
1.— Esta es sólo una interfase para discar y luego el usuario habla normalmente por el tubo.
2.— Esta interfase disca el número y se puede comunicar con otra computadora de la misma clase. ¿Cuál es la correcta?

FERNANDO L. VALLI
SAN JUSTO

K 64:

El razonamiento correcto es el primero. Esta interfase, junto con el soft que aparecía en la nota, sirve para discar los teléfonos que tengas en tu agenda en forma automática, repetir un número y cortar la comunicación.
Para poder comunicarte telefónicamente con otra computadora, necesitás un modem, y además un programa que no es el que se utiliza en este proyecto.

COMPATIBILIDAD

Quisiera saber si se puede pasar un programa de una máquina cualquiera a una C 64.

MARCELO ALONSO
TRES ARROYOS

K 64:

En teoría, esto siempre es posible. Pero cuanto más distinta sea la máquina original de la C 64, esta conversión se tornará cada vez más dificultosa, al punto de que puede ser más conveniente rehacer el programa y no traducirlo. Esto es especialmente cierto si el programa está escrito en código máquina, y los microprocesadores en cuestión son distintos.
Por otra parte, existe una limitación muy difícil de superar (aunque no imposible) si el programa original tiene una longitud que supera la memoria libre de la C 64

QUEJAS

Traten de colocar al dorso del cupón un aviso publicitario, para no tener que recortar un programa o algo de interés.
CARLOS VERUCCHI
OLAVARRIA

K 64:

A veces no todo es posible. Si no querés arruinar tu ejemplar, podés fotocopiar el cupón, o simplemente copiarlo en una hoja y enviar ésta.



Con cariño

Tal como me lo pidieron y se los prometí, aunque un poco con retraso les envío la fotografía en la que estoy intentando operar con la Commodore 64 que obtuve en el sorteo de Infocom, organizado por esa querida revista. Otra vez muchas gracias por la computadora y el cassette de juegos.
Un beso de Miguel Augusto Solari (Gualeguay - Entre Ríos).

CUESTIONES DE MSX

Quisiera consultarles lo siguiente:
1.— ¿Qué tan compatible es el sistema operativo MSX DOS de Talent con el MS-DOS de la PCs?
2.— ¿Es cierto que en assembler se puede disponer de 64 K para programar en la Talent DPC 200?
ADRIAN MALDONADO
LA RIOJA

K 64:

1.— La compatibilidad de ambos sistemas operativos está dada en lo que se refiere a lectura de archivos (grabados principalmente en ASCII). Esto quiere decir que un archivo de texto escrito en la IBM PC, puede ser leido y trabajado en la MSX, sin ningún tipo de limitación.
Si bien la MSX puede leer archivos de la PC, no puede ejecutar archivos de comandos, por ejemplo el dBase II, versión MS-DOS o el LOTUS 1-2-3. Esto se debe a que ambas máquinas tiene dos microprocesadores distintos, el de las PCs es de 16 bits y el de las MSX es de 8.
2.— Si bien podés disponer de los 64 K de RAM para tus programas en código máquina, esto implicaría trabajar en áreas de memoria que la máquina necesita para su funcionamiento. Si en tu programa se considera esta situación y se la soluciona, no hay problema en que uses toda la RAM para un programa.

CLUB Y DUDAS

Junto con el cupón para ingresar en el club K-64, les mando algunas preguntas:
1.— Quisiera saber ¿qué significa resetear el ordenador?
2.— Cuando compré la computadora hace tres meses, adquirí dos joysticks, que ya se me rompieron. ¿Me podrían decir cuál es el mejor joystick?
MARCELO PERALTA
ROSARIO

K 64:

Resetear una computadora significa que por medio de una tecla o interruptor, podés salir de cualquier programa o rutina (aunque la máquina se haya colgado) y retomar el control de la misma. Existen diferentes versiones de reset, pero la mayoría te destruyen el contenido de la memoria y es equivalente a encender y apagar la máquina.
Con respecto al joystick, todavía no se inventó ninguno que dure para siempre. Los hay mejores y peores (te podés guiar por los precios) pero dependiendo del uso (o abuso) que se haga del mismo, tarde o temprano llegarán al final de su vida útil.

MODEM DESORIENTADO

Desde hace tiempo deseaba tener un modem, y ahora que mi deseo está satisfecho, me encuentro con algunas dudas:
1.— No sé cómo conectarlo a mi máquina (una C 64).
2.— Tengo un ensamblador y monitor de código máquina, pero no sé realizar el programa para modem y me gustaría conseguir alguno. También me gustaría cartearme con chicos y chicas que tengan la C 64.
PABLO DIGNANI
MARCELO T. DE ALVEAR 2231
(1122) CAPITAL

K 64:

Si bien no nos decís de qué tipo de modem se trata, la mayoría de ellos se conectan en el port del usuario de la C 64. Podés verificar esto, dado que los conectores traseros de esta máquina son distintos.
Con respecto al programa, éste puede ser o muy simple o muy complejo. Si no te animás a hacerlo por vos mismo, te recomendamos que te dirijas a algún negocio de soft y preguntes si venden alguna copia de los programas que acompañan a los distintos modelos de modem. De todos modos, publicamos tu dirección completa por si algún lector tiene uno y te lo puede facilitar.

DESCIFRANDO LA C 64

Si bien no tengo computadora, me informo bien y cada vez aprendo nuevas cosas. Muchos amigos míos tienen Commodore 64, y quisiera hacerles dos preguntas:
1.— ¿Cómo se puede hacer, una vez cargado el programa y luego de jugar, para que aparezca el listado del mismo en la pantalla?
2.— La Commodore 64, ¿es realmente una buena computadora?
CRISTIAN PONTE
CAPITAL

K 64:

1.— Si el programa está escrito en basic, podés probar con un LIST. Si no pasa nada, debe tener algún tipo de traba anti-listado. Hay muchas formas de hacer esto, y corre por cuenta de tu habilidad el destrabarlo. Si el programa está escrito en código máquina, no podrás ver el listado del mismo a menos que poseas ciertos datos como ser la dirección de inicio del programa, y manejar un programa desensamblador a la perfección.
2.— Tantos cientos de miles de usuarios no pueden estar equivocados.

COLOR PARA CZ 1500

Antes que nada queremos felicitarlos por la revista, que de acuerdo a nuestro criterio es la mejor que existe en el mercado argentino.
Sólo tenemos una duda, que quisiéramos nos la aclaren: ¿Le podemos dar color a la CZ 1500? ¿Cómo?
CLAUDIO y JULIO TRONCO
ROSARIO

K 64:

Sí, se le puede dar color a la CZ 1500. La forma de implementarlo es bastante complicada, ya que deben rediseñar la sección de video de esta máquina, poner otro chip generador de video (un 6847) y otra ROM para manejar esta sección. Como proyecto de hard es interesante, aunque tal vez antieconómico (te conviene más vender la máquina y comprar una que tenga color).

Z-80 OPERANDO

Tengo una CZ 1000, y mis preguntas son:
1.— El procesaor Z-80 que utiliza la Commodore 128 ¿es el mismo que el de la CZ?
2.— Las instrucciones de un libro especializado en assembler son distintas a las del manual de CZ. ¿Funcionarán igual?
3.— ¿Qué diferencia hay entre la CZ 1000 y la TS 1000? ¿Qué diferencia hay entre la ZX 80 y la ZX 81?

SERGIO GULLONE
SANTIAGO DEL ESTERO

K 64

1. Sí, cuando la C-128 trabaja en modo CP-M, está activo el Z-80, que es el mismo que se utiliza en la CZ 1000, sólo que en ésta última funciona a mayor velocidad.

2.— Suponemos que el libro a que hacés referencia está dirigido al Z-80, y en tal caso las instrucciones deben ser las mismas, o en definitiva funcionan igual.

3.— Entre la CZ 1000 y la TS 1000 no hay diferencias desde el punto de vista del funcionamiento de ambas máquinas, aunque se pueden encontrar algunas pequeñas diferencias entre las ROMs de las mismas. Entre la ZX 80 y la ZX 81, existen bastantes diferencias. Por ejemplo, la ZX 80 tenía sólo 1 K de ROM, no manejaba decimales, y cuando se ejecutaba el programa se perdía la presentación de video. La ZX 81 es muy similar a la TS 1000.

A continuación brindamos la nómina de participantes que resultaron premiados en el último problema del mes que correspondió a nuestro N° 18 del mes de setiembre de este año.

LAPIZ OPTICO

Mottironi, Juan C.
CHACO
Dominguez, Germán
TUCUMAN

JOYSTICK PEEK

Bibiloni, José María
CARLOS CASARES
Vrech, Mario D.
ALTA GRACIA

SOFTWARE CIBERNE

Mellano, Andrés N.
CORDOBA
Ramirez, Osmar A.
ZARATE - BS. AS.
Cappellere, Marisa
LA PAMPA
Hebenstreit, Fernando J.
SANTA FE.
Stella, Pablo
C. BERMUDEZ
Martinez, César E.
CAPITAL
Braje, Norberto J.
ADROGUE - Bs. As.
Duque de Arce, Oscar
CORRIENTES
Compagnucci, Alejandro
MAR DEL PLATA - BS. AS.
Tatamoni, Gustavo R.
BECCAR - BS. - AS.

SOFTWARE GAME 48

Moschen, Daniel R.
AVELLANEDA - SANTA FE
Duarte, Anibal E.
SANTA FE
Plana, Alejandro M. A.
CAPITAL
Pirona, Norberto Rubén
SANTA FE
Rosas, Marcelo M.
AVELLANEDA - BS. AS.
Postigo, Miguel A.
TUCUMAN
Nardelli, Emilio H.
ROSARIO
Bonamino, Eduardo
LOS TOLDOS - BS. AS.
Muñiz, Sebastián J.
OLAVARRIA - BS. AS.
Palmeiro, Diego Ariel
CAPITAL FEDERAL

SOFTWARE PEEK

Detlefsen, Alejandro
BERAZATEGUI - BS. AS.
Frezza, Luis A.
V. BALLESTER - BS. AS.
Introcaso, Enrique A.
MONTE GRANDE - BS. AS.
Peveri, Claudio A.
ITUZAINGO - BS. AS.
DODE, Fernando
S. M. de TUCUMAN

HARD CONFUSO

Soy poseedor de una 2068, y he armado la interfase para utilizar los joysticks del número 14. Dado que la 2068 tiene un bus de direcciones activo alto, le he intercalado un inversor entre la máquina y la interfase que ustedes publicaron.
El problema es que al conectarla y encender la máquina, ésta se "cuelga".
Mi pregunta es: ¿hice lo correcto? ¿Cómo puedo hacer que la interfase funcione, si es que se puede?

GABRIEL FOGEL
OLIVOS

K 64:

No vemos el porqué de tu necesidad de agregar una compuerta inversora, ya que el bus de direcciones de la Spectrum también es activo alto.
La interfase publicada sirve tanto para Spectrum como para 2068, por lo tanto te recomendamos que pruebes sin la compuerta inversora, (suponemos que la habrás puesto en la línea 5, que es la única señal del bus de direcciones que se utiliza).

Si todavía no funciona, revisa cuidadosamente todas las conexiones de la misma, en busca de cortocircuitos, y finalmente verifica el correcto funcionamiento de los circuitos integrados.

BOLSA DEL USADO

(Publicación gratuita. Máximo: 20 palabras).

VENTO ATARI 2600 COMPLETO, + 1 cartucho de juego (combate). ₳ 250. Tratar Marcos Pernicone. TE: (0478) 2989 ó 3290.

EMULADOR SPECTRUM 100%, ₳ 30. Rodolfo Villanueva. T.E.: 601-3944. Guamini 5183, (1439) Capital.

VENDO TK 2000 color, en perfecto estado, con joystick y 15 juegos. Todo por ₳ 350. Diego Verruno, Ingeniero Mitre 562. Temperley Este (1834). T.E.: 244-6020.

VENDO COMMODORE 64, con disquetera 1541, impresora 802, Fast Load, caja disquettes acrílico, 20 discos, 30 cassettes, 50 programas utilitarios, con manual. Tres transformadores, y otros chiches por ₳ 1.200. Todo nuevo. Visitarme en Bahía Blanca y Lastra, Chascomús. Ángel Bruno.

VENDO CZ 1000 con dos cassettes. Manual en castellano y caja original. En perfecto estado. ₳ 60. Llamar al 52-1419. Pedir hablar con la familia Fazioli.

VENDO TALENT MSX con 4 juegos, cassette de demostración, manual, cables y caja, todo original. Llamar al 248-4509, o escribir a Monteagudo 328. Banfield (1828). Alejandro.

VENDO MODEM para TS 2068, CZ y TK. Norma Bell 103. Con Software de comunicaciones. Conexión directa. ₳ 200. T.E.: 021-210436.

VENDO MICRODRIVE + INTERFASE 1 + 4 CARTRIDGES con programas utilitarios. ₳ 200. T.E.: 021-210436.

VENDO CASIO KX101. Computerized Radio Cassette Keyboard. T.E.: 021-802325.


Papá, sabías
que a la Talent MSX
nada le es imposible?
para
jugar,
estudiar y
trabajar
Es cierto: a la computadora personal Talent MSX nada le es imposible, porque gracias a la norma internacional MSX* trasciende sus propios límites. Mientras que la mayoría de las computadoras de su tipo han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene ilimitadas posibilidades de desarrollo.
Todo lo que la computación pensará, ya está logrado en una Talent MSX.
Su éxito en el mundo de la enseñanza inteligente lo demuestra. Día a día más establecimientos educacionales dotan sus aulas con esta computadora personal con vocación y prestaciones de una grande.
Talent MSX tiene ilimitades posibilidades de desarrollo.
La computadora personal Talent MSX pone a su disposición un mundo de software y con la incorporación de sus periféricos dialoga de igual a igual con cualquier PC profesional.
Por todo esto Talent MSX es la futuro-compatible.
* MSX es marca registrada de Microsoft Corp.
dialogo - D.P.
Talent MSX
Inteligencia en crecimiento
Producida en San Luis por Telemática S.A.
Curso de introducción sin cargo y 6 meses de garantía.